Approximate Solutions for Multi-Server Queueing Systems with Erlangian Service Times and an Application to Air Traffic Management

by

Marcos Escobar Fernández de la Vega

S.M. in O.R., Massachusetts Institute of Technology (1996)
S.M. in E.E., Massachusetts Institute of Technology (1995)
B. S. in E.C.E., Instituto Tecnológico y de Estudios Superiores de Monterrey-CEM (1992)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

September 1998

Author ....
Department of Electrical Engineering and Computer Science
September 11, 1998

Certified by ....
Amedeo R. Odoni
T. Wilson Professor of Aeronautics and Astronautics and
Civil and Environmental Engineering
Thesis Supervisor

Certified by ....
Richard C. Larson
Professor of Electrical Engineering

Accepted by ....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Approximate Solutions for Multi-Server Queueing Systems with Erlangian Service Times and an Application to Air Traffic Management

by

Marcos Escobar Fernández de la Vega

Submitted to the
Department of Electrical Engineering and Computer Science
on September 11, 1998,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

This thesis is concerned with approximations of certain $M(t)/G(t)/n(t)/n(t)+q$ queueing systems. More specifically, we are interested in such systems under very general conditions such as time-varying demand and capacity, and high utilization, including occasional oversaturation. Conditions such as these cannot be addressed with existing methodologies.

We focus on $M(t)/G(t)/n(t)/n(t)+q$ systems that can be approximated fairly well by $M(t)/E_k(t)/n(t)/n(t)+q$ systems. The latter have a large number of system states, that increase with the system parameters k, n, q and the utilization ratio, and involve complicated state transition probabilities. We propose numerical methods to solve the corresponding Chapman-Kolmogorov equations, exactly and approximately

We first describe the exact solution technique of $M(t)/E_k(t)/n(t)/n(t)+q$ queueing systems. Then, we develop two heuristic solution techniques of $M(t)/E_k(t)/n(t)/n(t)+q$ queueing systems, and provide the corresponding complete state descriptions. We compare the exact and approximate results to validate our heuristics and to select the heuristic that best approximates the exact results in steady-state and under stationary conditions. We also propose two algorithms to vary the number of servers in the system, since many real-life problems involve such changes in response to variations in demand. Further results using our ELC heuristic show that our practical approach behaves well under nonstationary conditions, including varying capacity, and during the transient period to steady-state.

We conclude that our heuristic approach is an excellent alternative for studying and analyzing $M(t)/E_k(t)/n(t)/n(t)+q$ models and, as a by-product, many $M(t)/G(t)/n(t)/n(t)+q$ systems that arise in practice.

Finally, we present an application of the $M(t)/E_k(t)/n(t)/n(t)+q$ queueing model in the context of Air Traffic Management. This model appears to be a reasonable approach to estimating delays and congestion in an en-route sector in the air traffic system and can be used as an important building block in developing an analytical model of the entire Air Traffic Management system.

Thesis Supervisor: Amedeo R. Odoni
Title: T. Wilson Professor of Aeronautics and Astronautics and Civil and Environmental Engineering

Acknowledgments

First, thanks to my thesis supervisor, Professor Amedeo Odoni, for his support, trust, confidence, guidance, patience, time and valuable insights into Queueing Theory and Air Traffic Flow Management. It was a pleasure to work under his advice in these fascinating topics.

Thanks to Professor Emmy Roth for her valuable insights, continuous support and help, wonderful editing, and all the time she spent with me.

Thanks to Professors Richard Larson and John Tsitsiklis for being part of my doctoral committee.

Thanks to doctors David Lee, Peter Kostiuk and Dou Long from the Logistics Management Institute, for their support and insights into Air Traffic Flow Management.

Thanks to Wesley McDermott for his help with the computer programs.

Thanks to my friends from México, here and there, and my friends from around the World for their friendship and encouragement. I have had a wonderful time with them.

Thanks to my extended family, including my wife's family, for being there when I needed them.

Thanks to my parents and brothers, for their unlimited love and support.

And finally, thanks especially to my wonderful wife, Ana Bárbara, for her love, support, care and patience. This thesis and my work is dedicated to her.

The funds of this research were provided by the Logistics Management Institute and CONACYT-MEXICO (reference D210/916/92).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Objective

The motivation for this thesis is our desire to study, at least approximately, the behavior of certain $M(t)/G(t)/n(t)/n(t) + q$ queueing systems under very general conditions, including dynamic demand and capacity and periods when the utilization rate exceeds 1. This is not possible with the existing state of the art. While the $M(t)/G(t)/n$ system has been the focus of many studies, it still remains largely intractable. Many efforts have been devoted to obtaining approximations for the distribution of waiting times, the number of customers in the system, the queue length and busy periods. A thorough literature review of results for $M(t)/G(t)/n(t)/n(t)+q$ queueing systems, with special emphasis on systems with Erlangian distribution for the service times, is presented in Chapter 2. In our discussion of previous work in Queueing Theory, we also cover approaches for the analysis of the transient period and of systems with dynamic parameters.

In this research, we shall concentrate on $M(t)/G(t)/n(t)/n(t) + q$ systems which can be approximated reasonably well by $M(t)/E_k(t)/n(t)/n(t) + q$ systems and will develop numerical approaches for solving such queues under general dynamic (not *steady state*) conditions. While the queueing systems with Erlangian distributions of service times are considered "easy" in Queueing Theory, it turns out that, in practice, many difficulties arise because of (1) the very large number of system states that may be present with increasing Erlang order and increasing numbers of customers and servers and (2) the complex state transition probabilities that one has to consider.

Our general strategy will be to describe the exact solution technique of $M(t)/E_k(t)/n(t)/n(t)+q$ systems that contains a complete description of the system states in the $M(t)/E_(t)k/n(t)/n(t)+q$ system and to develop heuristic solution techniques for these systems that approximate the results obtained using the exact solution technique. The results obtained using the exact solution technique will be used to validate those obtained using our heuristic solution techniques. From this validation, we will select the heuristic solution technique that approximates best the exact results. We shall also develop two heuristic approaches to account for systems in which the number of servers changes over time. The importance of the feature of variable number of servers stems from its applicability: real-life problems involve variations in capacity in response to fluctuations in demand.

We will show that the heuristic of choice provides a computationally efficient and tractable way for approximating the exact, dynamic $M(t)/E_k(t)/n(t)/n(t)+q$ system and, by implication, many $M(t)/G(t)/n(t)/n(t)+q$ queueing systems which arise in practice.

We are also interested in the application of this system to Air Traffic Management. Therefore, a practical by-product of our work is the use of the $M(t)/E_k(t)/n/n+q$ queueing model, with variable number of servers, as a reasonably good model to estimate delays and congestion in an en-route sector in the air traffic system.

1.2 Organization and Outline

The organization and outline of this thesis is as follows. In Chapter 2, we present an extensive survey of the literature available for $M(t)/G(t)/n/n+q$ queueing systems, with particular emphasis on the case of systems with Erlangian service time distributions. We also cover various results for the analysis of the transient period, as well as various techniques for solving systems with nonstationary parameters. Most results in the available literature are concerned with steady-state solutions. We have identified various trends in research interests in Queueing Theory in this respect, between the late 1960's and the present. From the literature review, we conclude that few techniques can be used in modeling multi-server systems with general service times. This is especially true in the case of time-varying demand and capacity and periods when the system is over-saturated.

Chapter 3 addresses exact and heuristic solution techniques for the $M(t)/E_k(t)/n(t)/n(t)+q$ queueing system. This finite-capacity queueing system with time-varying Poisson arrivals and Erlangian service time distribution can be represented by a set of states for which we can write the Chapman-Kolmogorov equations. We present first the exact approach and describe the complexity of the state transitions. Then, we derive two heuristic solution techniques that simplify significantly the transitions among states. The fundamental idea in our heuristic approaches is the combination of multiple states in the exact representation into a single state in the approximate representation. What differentiates our two heuristic solution techniques is the algorithm to compute the state transition probabilities. In order to evaluate the performance of our heuristics, we describe several performance measures of interest, including aggregate probabilities and queue statistics.

Two other algorithms, one for the exact the other for the heuristic, to solve systems with a variable number of servers are developed in Chapter 3. The algorithms map the states of the system before the number of servers changes to the states in the modified system. The importance of those two algorithms is their wide applicability as many realistic problems involve time-varying capacities in response to changes in demand.

Chapter 4 validates our heuristic solution techniques in steady-state and under stationary conditions. We solve numerically the Chapman-Kolmogorov equations for both the exact and heuristic solution techniques and compare the results obtained. The comparison of the exact and approximate results has two objectives: validate the accuracy of the heuristics and select the heuristic of choice. The validation consists of a large set of conditions with a wide range of system parameters. We conclude that the heuristic ELC (Equally Likely Combinations) provides an excellent approximation to the exact results: 100% of the results using ELC are within 3% of the exact results and 95% are within 1%; the approximate results are computed up to 3 orders of magnitude faster than the exact results; and, larger systems that are impossible to solve with the exact solution technique, can be solved quickly using the heuristic solution technique.

In Chapter 4, we further analyze the performance of ELC under time-varying conditions and study the transient behavior of the $M(t)/E_k(t)/n/n+q$ model using both the exact and ELC solution techniques. The evidence in Chapter 4 suggests that our practical approach performs very well under the above circumstances. The results of the examples with variable number of servers show that the algorithms proposed capture reasonably well, at least intuitively, the system dynamics when the capacity of the model is either increased or reduced.

Chapter 5 describes the implementation of the solution techniques presented in Chapter 3 and includes a case study of an en-route sector in the Air Traffic System. The case study uses an $M(t)/E_3(t)/n(t)/n(t)+q$ queueing system to model the en-route sector and presents several scenarios with different demand and capacity patterns, including a baseline case with actual arrival data for a particular sector. We present this example to illustrate the potential applications of the $M(t)/E_k(t)/n(t)/n(t)+q$ queueing systems in modeling and analyzing some hypothetical questions about en-route sectors. The analysis shows that our model can be of great help in evaluating and planning daily en-route sector operations, and in assisting air traffic managers and administrators in developing strategies and policies to guarantee a satisfactory level of service and an acceptable workload for air traffic controllers.

Finally, Chapter 6 summarizes the conclusions and contributions of our research and briefly describes possible areas of future research.

We conclude this introduction with a remark found in Malone's thesis [27] that applies completely to the motivation and concerns of our work:

> *Concern for developing models to understand and analyze complex real-world dynamic systems motivates this research. Contributions are of both a quantita-*

tive and qualitative nature. Quantitatively, this research develops fast, accurate approximation methods for dynamic queueing systems of significant practical importance. These approximations are flexible and accurate, and it is hoped that they will be used as tools in future analyses. Qualitatively, we hope that the resulting improved understanding of complex dynamic queueing system behavior will provide rules of thumb to help planners and operators of facilities with strongly time-dependent demand and capacity to make better facility management decisions.

Chapter 2

Literature Review

The objective of this thesis is to study in detail the $M(t)/E_k(t)/n/n+q$ queueing system and to find exact and approximate solution techniques for such systems. We are also interested in the application of this system to Air Traffic Management. The aim of this Chapter is to present background material relevant to our research.

Our research was driven by the usefulness of the $M(t)/G(t)/n/n+q$ systems to model en-route sectors in the U. S. airspace, and possibly, to model independent runway systems in airports. The practical applications we are interested in include very general conditions: a wide range of utilization ratios, even over-saturated for some periods of time; variable capacity and demand; and multi-server systems. In this literature review we show that it has proven to be very difficult to solve, even approximately, such systems. This situation motivated us to analyze the behavior of certain $M(t)/G(t)/n/n+q$ systems that can be approximated reasonably well by $M(t)/E_k(t)/n/n+q$ systems and to develop numerical approaches for solving such systems under very general dynamic conditions.

Therefore, we have undertaken the task of reviewing methodologies to solve or approximate $M(t)/G(t)/n/n+q$ queueing systems, with particular emphasis on the case of Erlangian service times. The scope of our review also includes various methodologies to approximate time-dependent systems and several results for the analysis of the transient behavior of queueing systems.

The organization of this chapter is as follows. We have classified the literature reviewed into five main groups. The first group presents research on the $M(t)/E_k(t)/n/n+q$ queueing system, with either stationary or nonstationary parameters (see Table 2.1). This group is subdivided into sections for exact and approximate solutions. The second group presents methodologies to approximate the $M(t)/G(t)/n/n+q$ queueing systems, with constant or time-dependent parameters (see Table 2.2). This group is subdivided into three sections depending on the approach used to approximate the $M(t)/G(t)/n/n+q$ model under both stationary and nonstationary conditions. The third group shows results for the transient period of queueing systems, from start until the system achieves steady-state (see Table 2.3). Exact and approximate solution techniques for time-dependent queueing systems are presented in group four (see Table 2.4).

Tables 2.1 through 2.4 summarize the four groups described above. We now describe some of the symbols in Tables 2.1, 2.2, 2.3 and 2.4. In the column **Objective**, P_n means distribution of customers in the system, W refers to the expected waiting time in the system, L_q is the mean queue length, $P(Delay)$ is the probability of delay, P_i refers for the probability of i customers in the system, τ is the system time constant (time to reach steady-state), and "statistics" means that the objective was to obtain queue statistics, such as the mean number of customers in the queue, mean waiting time, etc.. In the column **Parameters**, we specify the particular conditions assumed, and if not specified, we assume $\rho < 1$, and any values for k (in Erlang distributions) and n. Under the column **Approach**, "Algebraic" means that the authors used algebraic manipulations to obtain their solutions; $M/(D,G)/(1,n)$ refers to the $M/D/1$, $M/G/1$, $M/D/n$ and/or $M/G/n$; and, "moments of G" indicates that the author used moments of the general distribution of service times in the queue. The rest of the symbols are self-explanatory. In Tables 2.1 through 2.4, we can observe the different trends of research and their evolution from the mid 1960's until the present. We will elaborate on these trends of research after we discuss the results found in the literature.

The last group of results presented in this chapter covers applications of queueing theory, as well as other methodologies to the modeling of different parts of the Air

Results for $M/E_k/n/n+q$ Queueing Systems

Solution Type	Objective	Parameters	Approach	Author, Year
Exact	P_n, state description	Stationary, $q = \infty$, $k = 2$	Algebraic	Shapiro, 1966
	P_n, state description	Stationary, $q = \infty$	Algebraic	Mayhugh & McCormick, 1968
	P_n, state description	Stationary, $q < \infty$, $n = 2$	Numerical	Murray & Kelton, 1988
	P_n, Number of States, state description	Nonstationary, $q < \infty$, $k = 3$ (solution), $\rho > 1$	Numerical	Lee, 1997
Approximate	W	Stationary, $q = \infty$	Use known results of $M/(M, E_k)/(1, n)$ systems	Maaløe, 1973
	P_n	Stationary, $q < \infty$, $k = 1, 2, 3, \infty$	Laplace Transform, residual time	Smith, 1987

Table 2.1: Results for $M/E_k/n/n+q$ Queueing Systems

Table 2.2: Approximate Results for $M/G/n/n+q$ Queueing Systems

Objective	Parameters	Approach	Author, Year
P_n	Nonstationary, $q<\infty$, $\rho>1$, $n=1$	Numerical, interpolate results of $M/(M,D)/1/1+q$ systems	Koopman, 1972
P_n	Nonstationary, $q<\infty$, $\rho>1$	Numerical, interpolate results of $M/(M,D)/n/n+q$ systems	Odoni & Kivestu 1976
P_n	Nonstationary, $q<\infty$, $\rho>1$	Numerical, interpolate results of $M/(M,D)/n/n+q$ systems	Kivestu, 1976
W	Stationary, $q=\infty$	Use W of $M/(M,D)/n/n+q$ systems and moments of G	Cosmetatos, 1976
W	Stationary, $q=\infty$	Use W of $M/(M,D)/n/n+q$ systems and moments of G	Takahashi, 1977
W	Stationary, $q=\infty$	Use W of $M/(M,D,G)/(1,n)/(1,n)+q$ systems and moments of G	Boxma, Cohen & Huffels, 1979
P_n	Stationary, $q=\infty$, $q<\infty$	Laplace Transform, residual time, use results of $M/G/(1,\infty)$ systems	Hokstad, 1978
P_n	Stationary, $q=\infty$, $q<\infty$	Residual time, use results of $M/G/(1,\infty)$ systems	Tijms, van Hoorn & Federgruen, 1981
L_q	Stationary, $q=\infty$	Laplace Transform, residual time, use results of $M/G/(1,\infty)$ systems	Ma & Mark, 1995
W, $P(Delay)$	Stationary, $q<\infty$	Residual time	Nozaki & Ross 1978
P_i, $i=n,\ldots,q$	Stationary, $q<\infty$	Laplace Transform, residual time	Miyazawa, 1986
P_n	Stationary, $q=\infty$	Diffusion algorithm	Kimura, 1983
P_n	Stationary, $q=\infty$	Diffusion algorithm and Hokstad's results	Yao, 1985
P_n	Stationary, $q<\infty$	Use PASTA, conservation law and approximation of W	Kimura, 1996

Table 2.3: Some Transient Results for Queues Using Numerical Solution Techniques

System	Objective	Parameters	Author, Year
$M/G/1/1+q$	P_n	Nonstationary, $q<\infty$	Koopman 1972
$M/(G,E_k)/1/1+q$	P_n, τ	Nonstationary, $q<\infty$	Kivestu 1976
$M/M/1/1+q$	P_n, τ	Stationary, $q<\infty$	Odoni & Roth, 1981
$M/M/n/n+q$	P_n, initial conditions	Stationary, $q<\infty$	Kelton & Law 1985
$M/E_k/n/n+q$	P_n, initial conditions	Stationary, $q<\infty$, $n=2$	Murray & Kelton 1988

Table 2.4: Some Results for Time-Dependent Queueing Systems

Approach	System	Objective	Author, Year
Diffusion algorithm	$M/G/1/1+q$	P_n	Newell, 1968
	Various	Survey	Kleinrock, 1975
Numerical solution	$M/G/1/1+q$	P_n	Koopman, 1972
	$M/G/n/n+q$	P_n	Odoni & Kivestu, 1976
	$M/E_k/1/1+q$	P_n, DELAYS	Kivestu, 1976
	$M/G/1/\infty$	P_n	Malone, 1995
	$M/E_{(3,k)}/n/n+q$	P_n, number of states	Lee, 1997
Stationary approximation	$M(t)/M/n/\infty$	Statistics	Green & Kolesar, 1991
	$M(t)/(M,G)/n/\infty$	Statistics	Whitt, 1991
	$M(t)/M/n/\infty$	Statistics	Green & Kolesar, 1993
Behavior analysis	$M/G/n/\infty$	Asymptotic behavior, stability	Heyman & Whitt, 1984
	$M(t)/M/n/\infty$	Degree of nonstationarity	Green, Kolesar, Svoronos, 1991

Traffic System. Most of the work has focussed on airport-related congestion. Less effort has been dedicated to understanding and modeling of congestion in the en-route sectors of the airspace. The lack of work addressing en-route congestion and delays played an important role in the motivation of our research.

We finish this chapter with a summary of the results reviewed. This summary includes a time-line of the results presented in this Chapter and of the leads we follow in our research.

2.1 Previous Results on $M(t)/E_k(t)/n/n+q$ Queueing Systems

Many methodologies have been used to analyze the stationary and nonstationary $M/G/n$ system. As obtaining an exact solution for such systems has proven to be mathematically intractable, most of the work has been focussed on obtaining approximations for the distribution of customers in the system and in the queue, the expected waiting time and the expected length of the queue. One special case for which exact solutions have been obtained, under steady-state and for the transient period to reach steady-state, is when the distribution of service times is Erlangian (see Kleinrock [17]). Although, there have not been obtained closed-form solutions for systems with Erlangian service time distributions. In Sections 2.1.1 and 2.1.2, we present exact and approximate solution techniques for the $M(t)/E_k(t)/n/n+q$ queueing systems, respectively.

2.1.1 Exact Solutions to $M(t)/E_k(t)/n/n+q$ Queueing Systems

In this Section, we present the available exact results for the $M/E_k/n/n+q$ queue. Shapiro [40] and Mayhugh and McCormick [28] generate an exact solution to stationary $M/E_k/n$ systems with unlimited queue size ($q = \infty$) by exploring the fact that the Erlang distribution of order k is a sum of k independent exponentially distributed random variables. Thus, a customer would need to clear k stages of exponential service before leaving the service facility. Using the method of stages (see Gross and Harris [8] or Kleinrock [17]), they fully characterized the system by writing the Chapman-Kolmogorov equations with all possible state transitions. As the Erlang order and the number of servers increase, the number of system equations grows rapidly.

Due to computer limitations, Shapiro in 1966 and Mayhugh and McCormick in 1968, solved the system equations by using algebraic manipulations that differ on a case by case basis. In both articles the system is solved in steady state and for unlimited queue size. Shapiro solved a special case with $k = 2$ and proposed a state description with two elements: the first element indicates the number of customers in the system and the second element indicates the number of customers in the second stage of service. Mayhugh and McCormick generalized Shapiro's results for any Erlang order k. Their state description consists of a $(k+1)$-tuple with the first element indicating the number of customers in the system and the subsequent k elements indicating the number of customers in the first through k^{th} stage of service. Neither Shapiro nor Mayhugh and McCormick provide closed-form solutions to the systems.

Rather, they only presented the ordinary differential equations describing the system and a method for solving the equations. The importance of Shapiro's and Mayhugh and McCormick's solutions is that they proved that a solution to the $M/E_k/n$ system exists and it is unique. An important contribution of their results is that the state description allows us to solve numerically systems with stationary and nonstationary parameters.

In 1988, Murray and Kelton [30] solved the $M/E_k/2/2+q$, with q large enough to effectively have an infinite capacity system, using a more detailed state description. In this case, the state is described by a two-element vector indicating the number of stages remaining in Server 1 and the total number of stages remaining in the system, including the customers waiting in the queue. Murray and Kelton differentiate between Server 1 and Server 2, and Server 1 is occupied first if both servers are idle. The transitions between states become rather complicated and it is difficult to extend their solution technique to more than 2 servers. A second consequence of differentiating among servers is that the number of states increases considerably, even for a small number of servers.

Much more recently, and unaware of the state descriptions proposed by Shapiro and Mayhugh and McCormick, Lee [21] in 1997 suggested a $(k+1)$-tuple state description that varies slightly from that of Mayhugh and McCormick. The main differences between those two state descriptions are that (i) Lee's state elements indicate the number of stages remaining in the facility instead of the number of stages already cleared, as in Mayhugh and McCormick; and, that (ii) Lee's state description indicates the number of customers waiting in the queue while Mayhugh and McCormick's indicates the number of customers in the system, including those being served. Both state descriptions are equivalent and generate the same number of states to represent the system.

Lee solved numerically a finite capacity $M/E_3/n/n+3$ system allowing him to analyze the system under stationary or nonstationary parameters with no restrictions in the utilization factor. He proved that the total number of states in the system (T_S), and Chapman-Kolmogorov equations to solve, is given by

$$T_S = \binom{n+k}{n} + q\binom{n+k-1}{n} \tag{2.1}$$

for any value of k, n and queue capacity q. The solution presented by Lee is only for the case when $k = 3$ and $q = 3$, although his state description and proof for the total number of states in the system are valid for any Erlang order k and any queue capacity q.

We generalized Lee's solution technique for any Erlang order k. A reason for using Lee's approach is that his methodology to obtain the state transitions in the system is clear. We present his methodology in Sections 3.1.2 and 3.1.3 for the general $M(t)/E_k(t)/n/n+q$ model.

An advantage of using numerical solution techniques, as used by Lee and us, over the solution techniques presented by Shapiro and Mayhugh and McCormick, is that we can analyze the system during the transient period to reach steady-state and under steady-state conditions, with constant or time-dependent parameters. The approaches by Shapiro and Mayhugh and McCormick are more complex both analytically and computationally, and can only be used under steady-state conditions and with stationary parameters.

Although many researchers have investigated the $M(t)/E_k(t)/n/n+q$ queueing model in general, we are unaware if any researcher has been able to provide closed-form expressions to solve such system.

2.1.2 Approximate Solutions of $M/E_k/n/n+q$ Queueing Systems

Approximate solutions for the $M/E_k/n/n+q$ queues have been presented by Maaløe [25] and Smith [42]. Maaløe, in 1973, obtained two heuristic formulae for the mean waiting time in queue, with unlimited queueing capacity and stationary parameters. His approximations are for steady-state. The first approximation uses the mean waiting time of the $M/E_k/1$ queue with an arrival rate of $(\frac{1}{n})^{th}$ of the arrival rate of the multi-server queue, i.e., $\lambda_{E_k,1} = \frac{\lambda_{E_k,n}}{n}$, and the approximate mean waiting time is given by

$$W_{E_k,n} = \frac{1}{n} W_{E_k,1}. \tag{2.2}$$

Maaløe showed that this approximation is very good for $k = 1$, the exponential case, when the utilization ratio tends to 1. Clearly, this approximation is poor when ρ is small and n is large since the probability that all servers are busy decreases as ρ tends to 0.

Maaløe's second approximation requires the mean waiting times of the $M/E_k/1$, $M/M/1$ and $M/M/n$ queues, for which exact results are known. Maaløe intuitively argues that

$$W_{E_k,n} = \frac{W_{M,n}}{W_{M,1}} W_{E_k,1} \tag{2.3}$$

is a better approximation for the mean waiting time than Equation 2.2, and that the use of $\frac{W_{M,n}}{W_{M,1}}$ compensates for the low traffic intensity. This is because the ratio $\frac{W_{M,n}}{W_{M,1}}$ is a function of ρ. We compare the approximation of Equation 2.3 with those suggested by Cosmetatos [4] and Boxma et al. [3], and present some numerical examples in Section 2.2.1 below.

An alternative way to solve the $M/E_k/n$ system approximately was presented by Smith [42]. Smith, in 1987, proposed an algorithm to compute the distribution of the number of customers in the system which turns out to be a direct implementation of Hokstad's [11] approximation for the special case of an Erlangian distribution of service times. The solutions obtained are only for steady-state and for a reduced set of values for the Erlang order: $k = 1$, 2, 3 and ∞ (in his examples, he assumed that $k = 100$ is large enough to approximate $k = \infty$). Smith does not provide any new insight into queueing theory developments and only presents a case study validating Hokstad's results with Erlangian service time distributions.

2.2 Previous Results on $M(t)/G(t)/n/n+q$ Queueing Systems

In this Section we present several techniques used to approximate general $M(t)/G(t)/n/n+q$ systems. We can classify the different methodologies in three main categories: approximations using known results of $M/M/n$ and $M/D/n$ systems; approximations using the residual time in service of customers in the system; and, approximations using diffusion algorithms.

2.2.1 Approximations Using Results for Systems with Exponential and Deterministic Service Times

The first approach approximates $M(t)/G(t)/n/n+q$ systems by using known results for $M(t)/G(t)/1$, $M(t)/M(t)/n$ or $M(t)/D(t)/n$ systems, or a combination of them. We present results approximating the distribution of customers in the system, as well as some results approximating particular queue statistics of the $M(t)/G(t)/n/n+q$ system.

Approximating the Distribution of Customers in the System

Along these lines, Koopman [20] and Odoni and Kivestu [36] suggested that for most applications the general service time distribution has a coefficient of variation somewhere in-between those of an exponential distribution and a deterministic distribution. Koopman, in 1972, analyzed the single server queue. Odoni and Kivestu in 1976 extended Koopman's work to multi-server systems. The Chapman-Kolmogorov equations describing the behavior of an $M(t)/M(t)/n$ system were solved using a Runge-Kutta method; they also solved numerically the differential equations for the $M(t)/D(t)/n$ system. The results for $M(t)/M(t)/n$ and $M(t)/D(t)/n$ systems provide the upper and lower bounds, respectively, for many $M(t)/G(t)/n$ queues. Odoni and Kivestu used a weighting formula to compute their results for $M(t)/G(t)/n$ systems. Kivestu [16] suggests that an alternative to interpolation is to use Erlang service time distributions for $M(t)/G(t)/n$ systems.

The importance of this body of work is that analyses can be done for the transient periods as well as for steady state. It also allows study of systems with nonstationary parameters. (Koopman presented an example with periodic service and arrival rates). Although queues are assumed to have finite capacity, the systems can be effectively infinite capacity if the queue size is large enough. The numerical approach allows analysis of queues with utilization ratios larger than one, a fact that is important because in many applications the system becomes over-saturated for some periods of time.

Approximating the Mean Waiting Time

In 1976, Cosmetatos [4] noted once again, that the mean waiting time in a system with general service time distribution, with coefficient of variation in the range $0 \leq C_s \leq 1$, lies between the mean waiting times of the $M/M/n$ and $M/D/n$ models with the same parameters. He used exact values for the waiting times of $M/M/n$ systems and approximations for the mean waiting time of $M/D/n$ systems, along with the first and second moments of the service distribution, to generate the $M/G/n$ results. The weighting function for combining the results of $M/M/n$ and $M/D/n$ systems is derived from the similarities between such systems and the $M/G/n$ queue, and is given by

$$W_{G,n} = \nu^2 W_{M,m} + (1 - \nu^2) W_{D,n}, \tag{2.4}$$

where $\nu^2 = (\beta_2 - \beta^2)/\beta^2$; and β, β_2 are the first and second moments, respectively, of the general service time distribution. Cosmetatos' results apply only in steady-state, for stationary systems with infinite capacity. Numerical results show that Cosmetatos' results are better than Maaløe's results, mainly in low traffic intensities (see Table 2.5 below).

Takahashi [44] in 1977 proposed an approximation that used the first and α^{th} moment of the service distribution instead of first and second moments. He argued that the second moment "is not suitable for estimating the mean waiting time of a multi-channel queueing system," and showed numerically that his results provide good approximations in low traffic intensities. His approach is somewhat different from that of Maaløe and Cosmetatos because he approximates the mean waiting time of the $M/G/n$ system with an expression that is a function of α and the mean waiting time of the $M/D/n$ model, and α is obtained from an expression depending on the mean waiting times of both $M/D/n$ and $M/M/n$ queues.

Boxma, Cohen and Huffels [3] in 1979 used Cosmetatos' idea of a weighting function in their approximation of the mean waiting time in an $M/G/n$ system. Boxma et al. defined two quantities: "cooperation coefficient" and "normed cooperation coefficient," and used them to capture the measure of cooperation among servers in the system. The cooperation coefficient $C_{G,n}$ in an $M/G/n$ queue is given by

$$C_{G,n} = \frac{W_{G,n}}{W_{G,1}} \leq 1, \tag{2.5}$$

and the normed cooperation coefficient $N_{G,n}$ in an $M/G/n$ system is

$$N_{G,n} = \frac{C_{M,n}}{C_{G,n}}. \tag{2.6}$$

The cooperation among servers refers to the difference between having multiple single-server systems, each of them with its own individual waiting queue, or having a multi-server system with a common waiting queue for all servers. In the former case, when a server becomes free, it may remain idle while there may be customers waiting in the individual queue of a busy server. Therefore, there is an advantage to having a multiple-server system since the expected waiting time in the queue may be smaller, especially with low utilization ratios. Boxma et al. used a weighting function similar to Equation 2.4 to approximate $N_{G,n}$. Then, using Equations 2.5 and 2.6 with the approximate value of $N_{G,n}$, Boxma et al. give the following approximation of the mean waiting time:

$$W_{G,n} = \frac{W_{M,n}}{N_{G,n} W_{M,1}} W_{G,1}. \tag{2.7}$$

If we let $N_G = 1$, we can obtain the same approximation for the mean waiting time

Table 2.5: Exact vs. Approximations of the Mean Waiting Time in an $M/E_4/3$ System

ρ	$W_{G,n}$ (Exact)	Maaløe (%)	Cosmetatos (%)	Boxma et al. (%)
0.1	0.00103	-16.85	2.18	-0.02
0.4	0.0536	-8.49	-0.37	-0.69
0.6	0.194	-4.71	-0.61	-0.66
0.8	0.688	-1.99	-0.40	-0.39
0.9	1.72	-0.92	-0.21	-0.20

suggested by Maaløe [25] for the $M/E_k/n$ queue. Note that Maaløe's approximation, Equation 2.3, assumes that the cooperation among servers is always the same, even at low traffic intensities.

In Table 2.5 we compare the exact mean waiting time in the $M/E_4/3$ system with the approximations of Maaløe, Cosmetatos and Boxma et al., for various utilization ratios. Columns 3 through 5 show the relative percentage errors. The values in Table 2.5 were obtained from Boxma et al. [3]. The exact results taken from Hillier and Lo [10], and double-checked using the exact solution technique described in Section 3.1.3. Notice that the Cosmetatos and Boxma et al. approximations perform significantly better than Maaløe's with low traffic intensities. Maaløe's approximation does not include the first and second moments of the service time distribution, and as expected, the performance of the approximation with low utilization ratios is poor because the "cooperation" among servers is not taken into account. Cosmetatos approximation improves over Maaløe's over all utilization ratios but Boxma et al.'s provides even better results for low utilization ratios.

These approximations are all heuristic nature and it is difficult to evaluate exactly the reasons of improvement of one over the other. We can see that these efforts to improve the approximations of the mean waiting time in the system take the form of adding more quantities like the moments of the service time distribution and the cooperation coefficients among servers, but no formal methodology has really been developed along these lines. All approximations of the mean waiting time presented in this section apply under stationary conditions and in steady-state.

2.2.2 Approximations Using Residual Times of Customers in the System

We present in this section two different approaches used to approximate $M/G/n$ systems using residual times. The first approach described is divided into two sections, depending on the number of customers present in the system, and uses known results for systems with infinite number of servers and single-server systems. The second approach obtains approximations for limited queue size systems.

Using Results for $M/G/\infty$ and $M/G/1$ Queueing Systems

The first approach uses residual times of customers in the system to approximate $M/G/n$ queues and separates the analysis of the queue into two parts. Then, results for the $M/G/\infty$ and $M/G/1$ systems are used in the approximation. Hokstad [11], Tijms, Van Hoorn and Federgruen [45] and, more recently, Ma and Mark [24] used this technique to derive their approximations. The analysis of the $M/G/n$ system is split into two parts. When the number of customers present in the system is below a certain threshold, the queue is assumed to behave like an $M/G/\infty$. Once the number of customers is above the threshold, the system is analyzed as an $M/G/1$ system with the mean service time scaled by the number of servers. The results presented below are for steady state of systems with stationary parameters and infinite queue size. The threshold for using $M/G/\infty$ or $M/G/1$ results differentiates the various approaches.

Hokstad, in 1978, considered that the $M/G/n$ queue behaves as an $M/G/\infty$ queue if the number of customers in the system m is less than the number of servers n, i.e., when $m < n$. For $m \geq n$, the results used are those of an $M/G/1$ queue with the modified service time. He used the Laplace transform of the service time distribution and the probabilities for the remaining times in service for all customers in the system to derive his approximations. He obtained an approximate distribution of customers in the system. Hokstad also extended his results to the finite queue size case and provided an expression for the total waiting time in queue.

In 1981, Tijms et al. proposed three different approximations for the $M/G/n$ queue in steady-state and under stationary conditions. They used the PASTA result (see Wolff [48]) and a recursive approach to generate the distribution for the number of customers in the system (m) in all three approximations. As a general approach for their approximations, at every epoch of service completion, they obtain the residual life for the smallest service time of the customers being served either using the results of an $M/G/\infty$ or $M/G/1$ queue. The first two approximations consider the same threshold as Hokstad in splitting the analysis of the queue, i.e., when $m < n$ or $m \geq n$. The only difference between those two approximations is that the second one simplifies one term inside of the recursion formula. The third approximation considers a special case for the situation when a customer leaves the system leaving all but one servers busy, i.e., only $n-1$ servers occupied, while all servers were busy when the leaving customer was in service. In the third approximation, Tijms et al. attempt to capture the transition from all servers busy to having at least one server idle with a variable threshold. Ma and Mark in 1995 followed the intuition from Tijms et al. that a variable threshold may provide a more accurate approximation.

Ma and Mark showed that their approximation, with the variable threshold, generated better results for low utilization ratios and for a large number of servers. Ma and Mark obtained the distribution of customers in the system, using the z-Transform approach, with the distribution of residual time in service and the distribution of the interdeparture time. They claim that the threshold for deciding which assumption to use depends on the number of servers, the utilization factor and the service distribution as those quantities determine the traffic intensity of the system. If the threshold is considered equal to $n-1$, where n is the number of servers in the system, Ma and Mark's results are exactly the same as those obtained by Tijms et al.'s first approximation. In the same article, Ma and Mark suggested that smaller systems, with the same utilization factor and same service distribution, can be used to approximate the mean queue length of larger systems. The problem is that they did not show a methodology to obtain the function relating both systems. They only provided certain characteristics for such functions and an example.

Table 2.6: Exact vs. Approximations of the Mean Queue Length for an $M/G/30$ Queue

ρ	L_q (Exact)	Tijms et al.	Ma and Mark	Miyazawa
0.10	5.325×10^{-21}	8.8427×10^{-21}	5.158×10^{-21}	5.518×10^{-21}
0.30	1.626×10^{-8}	3.807×10^{-8}	1.533×10^{-8}	2.113×10^{-8}
0.50	9.433×10^{-4}	2.007×10^{-3}	9.892×10^{-4}	1.279×10^{-3}
0.70	0.4158	0.6072	0.4369	0.4675
0.90	29.16	31.02	28.98	28.57
0.95	101.3	102.5	101.1	98.44

Approximations for Systems with Finite Queue Capacity

In a different approach to obtain steady-state approximations for stationary systems, Nozaki and Ross [33] and Miyazawa [29] used only the residual time and the service time distribution, along with the distribution of the number of customers in the system. Nozaki and Ross in 1978 and Miyazawa in 1986 assumed a finite queue capacity and allowed for over-saturated systems to be analyzed. Their numerical results compared favorably with those obtained by Maaløe, Hokstad and Tijms et al. The results from Nozaki and Ross and Miyazawa can be extended to infinite queue capacity. Miyazawa's results are better in lower traffic intensities than Tijms et al.'s results, as seen in Table 2.6.

Table 2.6 shows some examples comparing results for the mean queue length obtained by Tijms et al., Ma and Mark and Miyazawa, for an $M/G/30$ queue with Hyperexponential service time distribution. The values presented in Table 2.6 were taken from Ma and Mark [24], with the exact values computed by them. No information was presented on the technique used to calculate the exact results. In this reduced set of examples, we can see that there have been improvements in the approximations to handle lower traffic intensities and large number of servers, e.g. $n = 30$. Ma and Mark's results are better approximations than Miyazawa's and Tijms et al.'s results.

2.2.3 Diffusion Approximations of $M/G/n$ Queueing Systems

A third approach to approximate stationary $M/G/n$ systems in steady-state is by using diffusion approximations: a discrete queueing process, in this case the number of customers in the system, is approximated by a continuous diffusion process, e.g., $\{X(t)|X(t) \geq 0\}$. Kimura [13] and Yao [49] used this approach in their approximations.

Kimura in 1983, formulated a diffusion process approximating the number of customers in the system. The formulation assumes the infinitesimal mean and variance in the diffusion process as piecewise continuous functions. The model assumes that interdeparture times are independent, identically distributed random variables obtained from the service time distribution. Interdeparture intervals are independent of each other only when there are customers in the system. Otherwise, no departures can occur. Kimura assumed that when the process reaches zero, it stays there for an exponentially distributed time. We can see that a diffusion approximation tends to improve for high utilization ratios as the system will be in a busy period most of the time.

Kimura's approximation was improved in 1985 by Yao. He modified the boundary conditions and used Hokstad's result to simplify the diffusion process equations. The objective in Yao's modification is to obtain a diffusion approximation with better results in low traffic intensities. Both, Kimura and Yao, integrate the pdf of the diffusion process to obtain the distribution of customers in the system. In Table 2.7, we present results for an $M/E_2/10$ queue using Kimura's and Yao's approximations. The results were taken from Yao [49], with the exact values based on the tables of Hillier and Lo [10]. We compared the exact results with the values obtained using the algorithm in Section 3.1.3. Notice that Yao's approximation performs better tha Kimura's approximations in lower traffic intensities.

The last approach for approximating $M/G/n$ systems, presented in this Section, which we were unable to include in any of the other categories, is described as follows.

Table 2.7: Exact vs. Approximations of the Delay Probability and Mean Number of Customers in the System for an $M/E_2/10$ Queue

ρ	$P(Delay), L$ (Exact)	Yao	Kimura
0.10	0.11200×10^{-6} 1.0	0.11921×10^{-6} 1.0	0.00306×10^{-6} 1.012
0.30	0.11368×10^{-2} 3.0	0.11479×10^{-2} 3.0	0.025187×10^{-2} 3.008
0.50	0.0351 5.029	0.0359 5.028	0.0218 5.015
0.70	0.2166 7.407	0.2215 7.394	0.1955 7.267
0.90	0.6624 13.576	0.6686 13.521	0.6598 13.144
0.99	0.9627 81.547	0.9637 81.458	0.9631 80.937

Kimura [14], in 1996, presented a transform-free approximation for the stationary $M/G/n/n+q$ system in steady-state. He obtained the probability of saturation P_{n+q} using the PASTA result and a conservation law: the average rate of accepted arrivals is equal to the average departure rate, not including lost customers, i.e.,

$$\lambda(1 - P_{n+q}) = \mu E[\min(m, n)], \tag{2.8}$$

where m is the number of customers in the system. Kimura defined m_q and m_∞ as the number of customers in an $M/G/n/n+q$ and $M/G/n/\infty$ system, respectively, and obtained his approximation for the distribution of customers in the system assuming that m_q is related to m_∞ by truncating and renormalizing the distribution of m_∞. He suggested the conditioning approximation

$$\frac{P(m_q = j)}{P(m_q < n+q)} \approx \frac{P(m_\infty = j)}{P(m_\infty < n+q)}, \qquad j = 0, ..., n+q-1 \tag{2.9}$$

to generate his results. He extended his results to infinite capacity systems. One disadvantage in Kimura's approximation is that the distribution of customers in the system is a function of the mean waiting time in the system, which needs to be

Table 2.8: Exact vs. Approximations of the Mean Queue Length for an $M/E_3/3/3+10$ Queue

ρ	L_q (Exact)	ELC	Kimura	Miyazawa	N-R
0.3	0.02202	0.02184	0.02193	0.02001	0.02001
0.6	0.36752	0.36655	0.36348	0.35322	0.34017
0.9	2.74400	2.74405	2.70929	2.76944	0.84588

approximated as well. The advantage is that his approximation can be implemented relatively easier than previous approximations. In Table 2.8, we present numerical results for the mean queue length L_q in an $M/E_3/3/3+10$ queue with approximations of Kimura, Miyazawa, Nozaki and Ross, and our approximation using Heuristic 2, ELC, in Chapter 3. The exact results were obtained with the exact solution technique described in Section 3.1.3, and the results for Kimura, Miyazawa and Nozaki and Ross were taken from a table in Kimura [14]. Kimura used Boxma et al. approximations in obtaining his results. Notice that in general, the results are stably accurate except for Nozaki and Ross which becomes extremely poor as ρ increases. We included our ELC approximation to compare its performance with previous results, and to indicate that it compares favorably with those.

2.3 Research on the Transient Behavior of Some Queueing Systems

Queueing theory has focussed mainly on steady-state operations, when the effects of the initial conditions have faded out. Such analysis of systems may be inappropriate in many applied situations where the operations show periodic cycles or where operations finish at some point in time. A more appropriate analysis of such systems would be transient, where the system's operations are described for a fixed, finite amount of time and takes into account the initial conditions.

Our next step is to review results on transient analysis of queueing systems, with either stationary or nonstationary parameters. The solution techniques presented by

Koopman [20], Odoni and Kivestu [36] and Lee [21] allow for transient analysis of the queueing systems studied. Numerical solutions of exact systems provide results at each instant of time, with constant or dynamic parameters. With constant parameters, this approach helps to determine the time to reach steady-state, if it exists. In many real-life applications, we observe dynamic patterns of demand and service rates and it is important to investigate if the systems ever achieve steady-state or reach some equilibrium stage before the demand or service change, e.g., the hourly demand at an airport or en-route sector.

In the queueing literature, there are few papers addressing the transient phase of queueing systems. Kivestu [16], in 1976, investigated the behavior of the transient period of $M/M/1$ to understand that of an $M/G/1$ system. Along this line of research, Odoni and Roth [37] presented a detailed analysis of the transient behavior of single server queueing systems and tried to determine the time to reach steady state. They proved empirically that the time to steady state is dominated by an exponential factor with a time constant, defined as the relaxation time. This time constant is a function of the utilization factor among other parameters. Their analysis included infinite capacity, single server systems with stationary arrival and service rates.

In Section 2.1.1, we discussed the approach used by Murray and Kelton [30]. They presented a different transient analysis of the $M/E_k/2$ system. Their objective was to determine the effect of initial conditions on the transient behavior of the system, through the use of simulation techniques. Similarly, Kelton and Law [12] carried out a numerical examination to understand how the choice of initial conditions affects the convergence of expected delays to their steady-state values in $M/M/n$ systems with stationary parameters. Kelton and Law also discussed the implications of their results for the initialization of steady-state simulations.

Transient results are difficult to obtain and are available only for a restricted class of models, e.g., those that can be solved numerically. Many results of queueing systems involve transforms that are difficult and complicated to invert making it difficult or impossible to track the transients. Other results are expressed through complex functions that are difficult to evaluate for the transient period. Therefore, analysis of transients in queueing systems are not obtainable in many solution techniques.

2.4 Results on Queueing Systems with Nonstationary Parameters

The overwhelming majority of queueing papers to date has been dedicated to system with stationary parameters, as seen from the set of results presented in this Chapter, while many of the most interesting queueing problems in practice involve nonstationary parameters. In the queueing literature, we find two different principal approaches to solving time-dependent systems: numerical methods and stationary approximations of time-varying systems. We address both approaches and cite relevant methodologies in the two sections below.

An alternative methodology to approximate nonstationary systems was presented by Newell [31] in 1968. He used a diffusion algorithm similar to the one used by Kimura [13] and Yao [49], discussed in the previous section, but allowed for time-varying infinitesimal mean and variance. Newell used the diffusion approximation to analyze the behavior of single-server systems with slowly increasing arrival rate, which become over-saturated over a period of time (rush hour). Kleinrock [18] presents a detailed survey on diffusion approximations for systems with stationary and nonstationary parameters.

Another type of approach in studying nonstationary queueing systems includes asymptotic behavior analysis. For example, Heyman and Whitt [9], in 1984, analyzed the asymptotic behavior of queues with time-dependent arrival rates and presented some definitions of stability for systems with periodic and non-periodic arrival rates. An application of Heyman and Whitt's results is the study of dynamic steady-state associated with systems with periodic arrival rates.

Green, Kolesar and Svoronos [7] in 1991 also investigated numerically the behavior of multi-server systems with sinusoidal Poisson input. Green et al. showed that if systems that are even modestly nonstationary (e.g., the amplitude of the variability arrival rate is 10% of its average) are approximated with stationary models, the expected delays can be "seriously" underestimated. The importance of Green et al.'s findings is that, in most cases, the nonstationarity of queueing systems cannot be safely ignored without obtaining misleading results. Therefore, it is important to develop accurate (exact or approximate) solution techniques for time-dependent queueing systems.

2.4.1 Numerical Methods for Systems with Dynamic Parameters

Some of the results presented in this Chapter were obtained using numerical solution techniques. For example, Koopman [20], Odoni and Kivestu [36] and Lee [21], among others, solved numerically the ordinary differential equations of single- and multi-server systems with Exponential, Deterministic and Erlangian service time distributions. Although numerical solutions may be computationally expensive, they provide a reliable analysis of time-dependent queueing systems. Therefore, an interest in reducing the computational work involved in this type of approach has been the focus of many researchers.

In 1976, Kivestu [16] developed an algorithm (DELAYS) to approximate the $M(t)/E_k(t)/1/1+q$ system. His approach uses a set of differential equations similar to those of an $M(t)/D(t)/1$ queue, but the epochs at which the system is solved are scaled by a constant factor f. The epochs t_d, $d = 0, 1, \ldots$, are the times at which customers depart the system. In an $M(t)/D(t)/1$ system, there is a departure every $\frac{1}{\mu(t_d)}$ units of time. Therefore, Kivestu solved the modified differential equations every $\frac{f}{\mu(t_d)}$ units of time. The constant factor f depends on the time constants of the $M/E_k/1$ and $M/D/1$ stationary systems to reach steady state, e.g.,

$$f = \frac{\tau_{M/E_k/1}}{\tau_{M/D/1}} = \frac{k+1}{k}. \tag{2.10}$$

Notice that the modified epoch length is bounded above and below by

$$\frac{1}{\mu(t_d)} < t_d - t_{d-1} \leq \frac{2}{\mu(t_d)} \tag{2.11}$$

for values of $k = 1$ to $k = \infty$. The intuition is that scaling the epochs compensates for the time it takes a stationary $M/E_k/1$ system to reach the same state as a stationary $M/D/1$ system. Even though Kivestu solves the equations using the modified epochs, he obtains the probability of j arrivals to the system using the original epoch length, i.e., $\frac{1}{\mu(t_d)}$ units of time.

The modified system of equations is considerably smaller than the system of equations for an $M(t)/E_k(t)/1/1+q$ queue. In particular, there are $100 \times (1 - \frac{1}{k})\%$ fewer equations to solve. Also, notice that the modified time increment $t_d - t_{d-1}$ lies in-between the epoch lengths of the Deterministic and the Exponential service time distributions, as suggested in Koopman and Odoni and Kivestu.

More recently, in 1995, Malone [27] presented an approximation for the $M(t)/G(t)/1$ systems which does not assume any particular form of the service time distribution. The number of arrivals during the time a customer is in service is independent of the customer being served. Malone assumes that customer departures occur at t_d, $d = 0, 1, \ldots$, the *pseudo-departure* epochs, which allowed her to obtain the arrival rate just after a pseudo-departure epoch: $\lambda(t_d)$. She obtained a set of differential equations, exactly like those for an $M/G/1$ system, except that the probability for the number of arrivals during an interdeparture time has an explicit dependence on time. This dependence on time is reflected in the arrival rate $\lambda(t_d)$ and in the service time distribution at t_d.

An important contribution of Malone's approximation is that it works with any distribution of service time, and is not restricted to only Erlangian distributions, as Kivestu's approach. She validated her results with an extensive set of examples with various service time distributions for which exact or approximate solutions exist.

2.4.2 Stationary Approximations of Dynamic Systems

Several techniques use steady-state results of stationary systems to approximate solutions of nonstationary systems. Some of those techniques are the Pointwise Stationary Approximation, the Average Stationary Approximation, the long-run stationary approximation and the Modified-Offered-Load approximation.

The Pointwise Stationary Approximation (PSA) is a simple-to-use approximation since it only requires the steady-state expressions for systems with stationary parameters. PSA computes the approximations of long-run average performance measures with the arrival rate that corresponds to each point in time, i.e., we evaluate performance measures of systems in dynamic steady-state using the existing closed-form formulae of stationary systems, with arrival rate $\lambda = \lambda(t)$ at time t.

Green and Kolesar [5] in 1991 empirically proved that the PSA is an upper bound for the results in $M(t)/M/n$ systems with sinusoidal arrival rates. In Table 2.9 we present some examples comparing stationary, exact and PSA results. Stationary results are obtained by using the average arrival rate over the entire period using the stationary $M/M/n$ model. The results presented in Table 2.9 were extracted from Green and Kolesar [5]. The exact delays resulted from numerical integration of the Chapman-Kolmogorov equations of the system. As shown in Table 2.9, there are cases in which PSA does not provide an accurate approximation. Green and Kolesar proved with examples that as the frequency of events increases, the performance of PSA also improves. Green and Kolesar presented a sensitivity analysis of the accuracy of PSA as the number of servers, and the arrival and service rates vary. They observed that (1) as ρ increases, with a low service rate μ, PSA deteriorates in performance; (2) as μ increases, PSA improves its performance even for high utilization ratios; and, (3) PSA outperforms the stationary approximation as the number of servers increases.

Whitt [46] also in 1991 proved that PSA is asymptotically correct as the arrival and service rates increase, while the traffic intensity remains constant. Whitt suggested that an intermediate approximation, the Average Stationary Approximation (ASA) may be useful for the analysis of $M(t)/G/n$ systems. The ASA method uses averages

Table 2.9: Comparing Exact Results with Stationary and PSA Results for an $M(t)/M/n$ Queue with Relative Amplitude = $\frac{A}{\lambda}$ = 1 (A = amplitude of sinusoidal arrival rate)

$\lambda = 6, \mu = 2$			
n	$P(Delay)$ (Exact) $E[Delay]$ (Exact)	Stationary	PSA
6	0.4851 0.2539	0.0991 0.0165	0.5446 —
9	0.0860 0.0125	0.0040 0.0003	0.0888 0.0132
12	0.0084 0.0007	0.0001 0.0000	0.0087 0.0007
$\lambda = 1, \mu = 0.2$			
n	$P(Delay)$ (Exact) $E[Delay]$ (Exact)	Stationary	PSA
9	0.1214 0.1740	0.0805 0.1006	0.2298 0.5962
12	0.0149 0.0122	0.0059 0.0042	0.0365 0.0373

of arrival and service rates in an interval of time, and obtains results also using the steady-state expressions for systems with stationary parameters. The length of the time interval is proportional to the mean service time. For example, if the utilization ratio exceeds one for a short period of time, but the time-averaged utilization ratio remains under one, then ASA can be used while PSA cannot. The PSA method cannot be used when the traffic intensity is close to one or above it because closed-form expressions for stationary systems do not exist. The difficulty of extending the use of PSA and ASA to the $M(t)/E_K(t)/n$ or $M(t)/E_K(t)/n/n+q$ systems is the lack of closed-form formulae to use at every integration step. An extension of PSA is the Simple Peak Hour Approximation (SPHA). SPHA uses the average arrival rates during the peak hour and obtains the expected peak hour expected delay, expected queue length and delay probability. The SPHA was presented in 1993 by Green and Kolesar [6].

2.5 Application of Queueing Systems in Air Traffic Management

In this Section, we describe models developed to measure capacity and delays in the airports and en-route sectors of the Air Traffic System. We classify the models by the methodology used in analyzing the models: analytical or simulation.

Even though we do not address simulation-based results in our research, we list and describe briefly existing models that use simulations. We also present a few analytical models that do not include the use of queueing theory but are relevant to the application discussed in Chapter 5.

The discussion of the models below is a summary from the NASA/AATT report presented by Odoni et al. [35], with the exception of the descriptions of LMINET and the enhanced AND model.

2.5.1 Analytical Models

Two types of models using analytical methods to obtain results are described in this Section. We have those focussed on airport capacities and those dedicated to computing delays.

Airport Capacity Models

We begin by describing two airport capacity models: the FAA Airfield Capacity Model, and the LMI Runway Capacity Model. The FAA model calculates the capacity of a runway system with continuous demand. It models a system of runway configurations, from single-runway operations up to four active runways, for a total of 15 different configurations. The FAA model assumes that each of the 15 configurations can be viewed as a combination of four basic configurations: single-runway, closely-spaced parallel runways, intermediate-spaced parallel runways and intersecting runways. For each runway configuration, the model computes the "all arrivals" capacity, the "all departures" capacity and the capacity of mixing arrivals and departures without reducing the arrival capacity. Once the above capacities have been calculated, the model interpolates to obtain runway capacities with different mixtures of arrivals and departures.

The FAA model implicitly assumes that taxiways and gates do not affect considerably the airfield capacity. This model can be used in policy-level analysis with quick approximate estimates of airfield capacities with varied parameters. A weakness of the model resides in the logic used to insert departures between arrivals on a runway causing misleading estimates when the number of arrivals is similar to the number of departures.

The Logistics Management Institute (LMI) Runway Capacity Model attempts to account for the stochasticity of airport operations by using normal random variables to model input variables such as approach speed and runway occupancy time. The model uses a "controller-based" point of view in spacing aircraft in the approach path. An important result in the model is the "runway capacity curve," including four basic points: "all arrivals," i.e., the runway is dedicated exclusively for arriving aircraft; "freely inserted departures," with the same number of arrivals as in "all arrivals" but some departures are inserted in-between arrivals; "alternating arrivals and departures," with an equal mix of arrivals and departures; and, "all departures," for a runway dedicated to departures only. Other arrival and departure mixes can be obtained by interpolating the points above. At this point, the LMI model is in its development stage and only obtains capacities for single-runway airports and add some ad hoc extensions to model multi-runway airports.

As suggested in [35], a mixture of the LMI Runway Capacity Model and the FAA Airfield Capacity Model, including the logic used in the LMI model with the multiple-runway configurations in the FAA model, "could be a very useful tool that would provide instantaneous estimates of runway system capacity with limited data requirements."

Delay Models

The analytical models presented in this Section use queueing systems to estimate delays at airports and en-route sectors in the air traffic system. The first using queueing models to investigate delays and saturation probabilities at airports was that of by Koopman [20], discussed in Section 2.2.1. Koopman presented several interesting example for JFK and La Guardia airports in New York with different capacities and computed the expected delays at the airports. Odoni and Kivestu [36] presented a handbook for estimating the average daily minutes of delay at major airports. The handbook contains several demand profiles that can be matched with the actual demand profile of an airport to estimate the total daily delay minutes.

The DELAYS algorithm, developed by Kivestu [16], computes the average delay for all operations in a runway without differentiating between arrivals and departures. DELAYS can be used for policy-oriented studies for obtaining approximate delay costs and compare the performance of different alternatives to improve or expand airports, or assess the efficiency in managing the existing demand at a particular airport.

DELAYS is also used as the queueing engine in computing delays at airports in the Approximate Network Delays (AND) model. The AND model consists of a network of airports, which are represented as interconnected queues. AND's objective is to analyze the impact of changes in airline schedules, traffic volume and airport capacity on flight delays on a national or regional basis. AND requires as input the capacity and demand profiles, which are given in hourly data that can vary from hour to hour. It also needs the flight schedule between all airports to be modeled and the detailed itineraries of the aircraft performing the scheduled flights. The current version of AND includes the 58 busiest airports in the United States. The output provided by AND are the hourly expected queue length at each airport, the hourly expected waiting time per operation at each airport, the total delay suffered by aircraft during the entire period of interest at each airport, and the fraction of aircraft delayed more than certain amount of time at each airport. The statistics above are computed with the probability vector $P(i,t,k)$, which is the probability that i aircraft will be in queue at time t in airport k.

The AND model does not account for congestion in en-route sectors. Some of the assumptions in the model are as follows: the airports (queues) are "weakly" connected, which means that no airport receives more than approximately 25% of its flights from any other single airport (which is true for practically all major commercial airports in the world); the operations are not distinguished as by arrivals and departures and aircraft are served in a first-come-first-served discipline, but the variations in the traffic mix can be adjusted in the hourly capacity at each airport. AND also assumes "that the delay suffered by each airport operation is equal to the expected value of the delay at the time when that operation is scheduled to take place." A key feature of AND is the propagation of delays through the network of airports. A detailed description of AND is presented in Malone [26] and [27].

In a parallel research project to the one presented in this thesis, an enhanced version of AND has been started. The enhanced AND model includes several en-route sectors that are modeled as multi-server queues. At this point, the enhanced AND model is in its initial stage of development. Two of the fundamental problems to overcome are the limited queue size in the en-route sectors and the rejection and re-routing of aircraft that intend to cross highly congested en-route sectors.

LMI has also developed a queueing network model (LMINET) for the United States airspace. LMINET models flights among a set of airports that traverse en-route sectors. The inputs to the model are the sequencing of en-route sectors, the airport capacities (given by the LMI Runway Capacity Model), schedules of arrivals and departures of its airports and weather information. The current LMINET models 64 of the busiest airports in the United States. The sectors are assumed to be rectangular square with roughly 120 miles on a side. Highly congested sectors are divided in subsectors to facilitate the queueing calculations. The sector division is performed empirically to reduce the substantial delays in the sector.

The development of the enhanced AND model and LMINET has been done simultaneously, with constant interactions between the MIT and the LMI research groups. For example, we influenced the modeling of the LMINET en-sectors with an $M/E_k/n/n+q$ queueing system instead of an $M/D/n/n+q$ queueing system; the enhanced AND model uses the same rectangular sectors and subsectors used in LMINET, and follow the same trajectories between airports. Currently, LMI is trying to implement $M/E_k/1$ queues to model their airports, following the same approach used in AND. Two important differences between the enhanced AND model and LMINET are that LMINET does not propagate delays and LMINET does not provide flight and airframe specific information. LMINET computes only hourly aggregate statistics for airports and en-route sectors. When LMINET reports large delays and high congestion at airports, the flight schedules are modified in a similar method as the FAA's current practice: scheduled aircraft departures to congested airports are delayed. Aircraft that already departed to the congested airport cannot be delayed, and flights coming from airports other than the 64 modeled by LMINET cannot be delayed either. A description of the latest version of LMINET is presented in Lee et al. [23].

2.5.2 Simulation Models

We classify the simulation models presented in this Section by the type of simulation used: deterministic, node-link (N-L) and 3-Dimensional (3D) simulations. The only model using deterministic simulation is the National Airspace System Performance Capability (NASPAC) model. NASPAC objectives are to obtain statistical reports of delays and flow rates. It was first conceived to undergo studies of strategic analysis of national airport investments but has evolved to provide analysis of tactical nature. NASPAC models airport runways and terminal and en-route airspace and uses as input the flight schedules and the constant airport capacities. It is a low-level-of-detail simulation model. One important feature is the itinerary generator. The itinerary generator infers the flight legs a particular aircraft follows through out the day. On the other side, NASPAC has long turn around times, is expensive and has questionable validity (mainly due to the constant airport capacity assumption). It's use requires extensive training and is labor intensive.

Node-Link Simulation Models

Node-link simulation models discretize airports and airspace into nodes and links. Conflicts occur when two or more aircraft try to move to a node using the same link. The conflicts are resolved by delaying one or more aircraft at a node according to a pre-programmed strategy. Three models use N-L simulation: The Airport Machine, SIMMOD and FLOWSIM.

The Airport Machine simulates in detail runways, taxiways and apron areas in airports. The model covers all aircraft from a few minutes before landing until a few minutes after take-off. It measures the flows and throughput capacities on the airfield per unit of time, and provides the delays incurred at each airfield facility. The Airport Machine relies on high-level-of-detail network representation of airfields where planes move along the network of links and nodes. One assumption is that take-off operations are independent from the route that the aircraft will follow after take-off. The model simulates one airport at a time. The Airport Machine can be used for design-level studies and to evaluate airport capacity and delays. In order to use the model, the user needs to undergo a significant amount of training and it is also considered labor intensive.

SIMMOD can be used with multiple airports at a time. The objective is to measure aircraft travel times, flows and throughput capacities. SIMMOD uses Dijkstra's shortest path algorithm to determine aircraft paths when they are not pre-specified by the user. Aircraft moves along a pre-specified high-level-of-detail network with also specified "rules of the road." It has many options for simulating probabilistic events and provides highly detailed output statistics. Although the user interface is poor. SIMMOD requires considerable training, the (expert) user must also be knowledgeable in ATM concepts and procedures, and it is labor intensive.

The Airport Machine and SIMMOD are widely used. The former is less labor intensive than the latter. In order to choose either simulation model, there are few trade-offs to consider in the decision: cost, quality of user interface, and model features and flexibility.

In FLOWSIM, the objective is to obtain delays and "ripple" effects induced by capacity constraints. Such constraints are given by the airport capacities since en-route sectors are assumed to have unlimited capacity. The delays are calculated using airport capacity models and the miles-in-trail restrictions. FLOWSIM models airport runways and terminal and en-route airspace. In this model, aircraft fly through pre-specified flight plans. FLOWSIM is the first prototype and is still under development. The current version is simple and allows for some user interaction. Comparing FLOWSIM and NASPAC, both model the same areas of the ATM system but FLOWSIM is faster in operation than NASPAC.

Three-Dimensional Simulation Models

In this type of simulation models, aircraft are allowed to fly 3D routes in the airspace with either pre-specified flight plans or flight paths that are derived from solving aircraft dynamic equations. In the latter case, aircraft dynamic equations are solved to simulate aircraft performance, causing actual flight plans to vary from the original flight plans specified by the user. When airplanes are on the airport surface, the simulation becomes two-dimensional. We present four models using 3D simulation: TAAM, HERMES, TMAC and ASCENT.

The Total Airspace and Airport Modeler (TAAM) is a very comprehensive simulation model: covers the complete gate-to-gate ATM process in detail. It can be used as a planning tool or to conduct analysis and feasibility studies of ATM concepts. TAAM is a high-level-of-detail model which requires an extensive training and it is labor intensive. It provides many options and flexibility and has an excellent interactive graphic user interface. TAAM cannot model dynamically special airspace use or hazardous weather, and it may not resolve all conflicts encountered. TAAM is the most fully featured ATM simulation tool, including dynamic re-routing of airplanes, but it is considerably expensive.

SIMMOD and TAAM are also competitors. Again, it is important to evaluate the trade-offs for each model, as described in the preceding section, to wisely select the use of any of them. Along with The Airport Machine and SIMMOD, TAAM is also a widely used simulation tool.

Another 3D simulation model is the HEuristic Runway Movement Event Simulation (HERMES). The objective in HERMES is to evaluate parallel runway capacities and operations, and provides average delays of aircraft using the runways. The model is tailored specifically to represent the operations of London Heathrow and London Gatwick airports. Real flight data is used in the model and it yields accurate results. It has not been used widely because it is difficult to generalize to any other airport. HERMES is labor intensive and requires an expert user.

TMAC objective is to determine conflicts and delays in the ATM system. It does not have the capability to resolve the conflicts. A nice feature of the model is that captures uncertainties of the trajectories, very useful in analyzing concepts such as Free-Flight. The en-route sectors in the model are assumed to have infinite capacity. TMAC is a high-level-of-detail, complex multi-element simulation model, which is intended to solve specific problems. It is not used as a generic modeling tool.

Finally, the last model presented in this review is ASCENT. ASCENT evaluates system-wide impact of new procedures, technologies and improved infrastructure. It covers all activities in the ATM process: airports, strategies (ground holds), weather, and en-route airspace, among others. The main focus of the model is in the terminal area operations. ASCENT is easy to use and allows for user interaction with a good graphical user interface. This model is recent and has not been validated adequately yet.

ASCENT, TMAC, NASPAC and FLOWSIM can be used to simulate airport runways and airspace. NASPAC and FLOWSIM are simpler to use than ASCENT and TMAC, which are still in early stages of development. NASPAC is the more mature model and is followed by FLOWSIM (which is also a prototype).

An advantage of 3D simulation models over N-L simulation models is the flexibility provided in analyzing aircraft in the airspace. N-L cannot be used to simulate the effects of implementing Free-Flight.

2.6 Summary

In the previous sections, we have identified and reviewed trends in the research of multi-server queueing systems, with both stationary and nonstationary parameters. We summarize the results presented in Table 2.10. The purpose of this Table is to illustrate the shifts in emphasis over the years. As seen in Table 2.10, early work was focussed on finding exact solutions to the $M/E_k/n$ models (late 1960's) through solving a large number of equations.

The next trend of research focussed on approximations to the more general $M/G/n$ queue. Most developments in this respect took place in the 1970's and 1980's. Interestingly, many of the approximate solutions to systems with general service time distributions used the Erlang distribution as an example in their numerical results. This suggests that $M/E_k/n/n+q$ systems may be used to approximate reasonably well certain $M/G/n/n+q$ systems. The validation approach used by several authors included the numerical comparison of their results with the well known results for $M/M/n$ and $M/D/n$ queues, as well as with previous work in the same area. The approximations to $M/G/n$ systems presented, and the improvements suggested by some authors to previous results, are of a heuristic nature. Consequently, it is difficult to validate the techniques used to obtain the approximations as no theoretical methodologies were used. In some cases, extra parameters were added intuitively, showing certain degree of improvement when comparing the new results with previous ones.

Most of the results presented in this review are concerned with steady-state solutions. Interest in the analysis of the transient behavior of queues grew in the early 1980's. With more computing power, larger and more complex systems could be considered. Even though there was an early interest in analyzing time-dependent systems, it was not until the late 1980's and 1990's that more techniques were developed to approximate such systems.

The analysis presented in this Chapter confirms that few techniques can be used in modeling the type of applications that motivated our research. Those applications

Table 2.10: Literature Review Summary

Year	$M/E_k/n$	$M/G/n$	Transient	Dynamic
1966	Shapiro			
1968	Mayhugh & McCormick			Newell
1972		Koopman	Koopman	Koopman
1973	Maaløe			
1976		Odoni & Kivestu, Kivestu, Cosmetatos	Kivestu	Odoni & Kivestu, Kivestu
1977		Takahashi		
1978		Nozaki & Ross, Hokstad		
1979		Boxma, Cohen & Huffels		
1981		Tijms, van Hoorn & Federgruen	Odoni & Roth	
1983		Kimura		
1984				Heyman & Whitt
1985		Yao	Kelton & Law	
1986		Miyazawa		
1987	Smith			
1988	Murray & Kelton		Murray & Kelton	
1991				Green, Kolesar & Svoronos, Green & Kolesar, Whitt
1993				Green & Kolesar
1995		Ma & Mark		Malone
1996		Kimura		
1997	Lee			
1998	Escobar, Odoni & Roth		Escobar, Odoni & Roth	Escobar, Odoni & Roth

require a wide range of utilization ratios, even over-saturated for some periods of time; variable capacity and demand; and multi-server systems. For example, Lee's results, that were motivated by our research, have been used to approximate the ATM applications of interest. The problem with Lee's approach is that it is computationally expensive. Other examples include the simulation models described in Section 2.5. Such models are even more computationally expensive and implementation requires considerable training and expert users. A disadvantage of using simulation methods over analytical ones is that many experiments need to be performed to obtain meaningful results.

We have followed some of the leads that were suggested in the analysis of queueing systems. From earlier work in $M(t)/E_k(t)/n/n+q$ queues, we used the method of stages to propose a new state description and suggested a heuristic technique to reduce the number of Chapman-Kolmogorov equations to solve. The numerical solution technique used to solve the reduced set of ordinary differential equations was influenced by the results of Koopman, Odoni and Kivestu and Malone. We also provided a time-varying solution of the Chapman-Kolmogorov equations, as Koopman, Odoni and Kivestu and Malone did as well. With our heuristic, we can also perform a transient analysis similar to the one presented by Odoni and Roth to determine the time-constants of $M(t)/E_k(t)/n/n+q$ models.

In the following chapters, we describe thoroughly our approach in obtaining exact and approximate solutions of $M(t)/E_k(t)/n/n+q$ queueing systems. We will also see that our heuristic performs well under nonstationary conditions and various utilization ratios allowing us to use it with the applications that motivated our work.

Chapter 3

The $M(t)/E_k(t)/n$ and $M(t)/E_k(t)/n/n+q$ Queueing Systems

In this Chapter, we describe in detail how to obtain exact solutions for the $M(t)/E_k(t)/n$ and $M(t)/E_k(t)/n/n+q$ queueing systems. We also introduce four heuristic techniques which simplify greatly the solution complexity.

We start by describing the method of stages used to represent the Erlang distribution in the $M(t)/E_k(t)/1$ system to enable solution of such systems. Then, we extend this approach to the case of multiple servers with limited and unlimited queue size. Finally, we address the case with variable number of servers. Note that all the results in this chapter apply under both stationary and non-stationary conditions, unless otherwise specified.

3.1 $M(t)/E_k(t)/n$ and $M(t)/E_k(t)/n/n+q$ Solutions

Systems with Poisson arrivals and Erlangian service time distributions can be characterize by a finite-state Markov process and a set of transitions among the states. In the following sections, we provide the state description for single- and multi-server systems with Erlangian distributions for the service time.

3.1.1 State Description for Single-Server Systems

In the $M(t)/E_k(t)/1$ queue, the service time can be interpreted as each customer needing to complete k independent, exponentially distributed sequential stages before leaving the service facility. Thus, each customer can be considered as a *package* of k tasks to be performed by the service facility. The service rate for each stage is $k\mu(t)$, with a corresponding expected time of $\frac{1}{k\mu(t)}$ per stage. The expected service time for completing all stages is $\frac{1}{\mu(t)}$.

The usefulness of this approach is that we can derive a state transition diagram with independent, exponentially distributed transitions that completely describes the queue. Due to the memoryless property of Poisson processes, in any time increment δt, the state can change only as indicated in the diagram. In the $M(t)/E_k(t)/1$ system, the states are defined fully by the total number of stages (or tasks) remaining in the system to be completed for all customers. Figure 3-1 shows the state transition diagram for this queue. For clarity, the time dependence has been omitted in the figure. Additional details are available in [17].

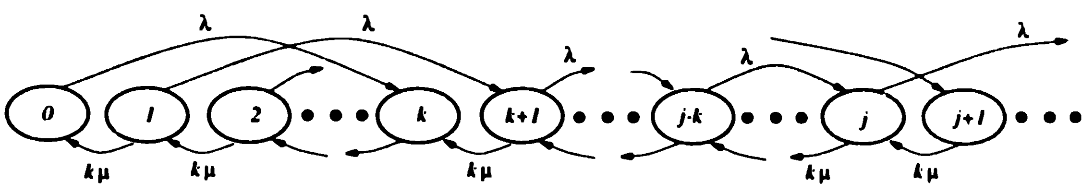


Figure 3-1: $M(t)/E_k(t)/1$ Queue

Using Figure 3-1, we can derive the Chapman-Kolmogorov equations describing the behavior of the number of stages and, therefore, the number of customers in the system over time. If the Chapman-Kolmogorov equations can be solved, performance measures such as queue length and expected waiting time can be obtained.

3.1.2 State Description for Multi-Server Systems

If the system has multiple servers, more information is needed to characterize completely the system state. The total number of stages is not sufficient to describe the state of the system because, for a particular number of stages, the distribution of such stages among the servers may not be unique. In this section, we illustrate this complication in greater detail, specify additional information required and show how to obtain the exact state probabilities. We also introduce two heuristic approaches to obtain the probability distribution of the number of customers in the system.

The information needed to describe the states in a multiple server queue with exponential interarrival times and Erlangian service times, is the following:

- Number of uncompleted stages in the system
- Number of customers in service and in the queue
- Distribution of uncompleted stages among the busy servers

There are at least three ways to describe the states in the $M(t)/E_k(t)/n$ queue. First, let the system state be of the form $(x_1, x_2, ..., x_n, a_q)$ where x_i indicates the number of uncompleted stages at server i, $1 \leq i \leq n$, and a_q indicates the number of customers waiting for service in the queue. This enumeration is a finest grain description for the system state. Because of the extremely large number of states, qn^k, this description is not considered further in our work.

The second way to describe the states is the one suggested by Shapiro [40] and Mayhugh and McCormick [28], and similarly by Lee [21] (see Section 2.1.1 for specific differences among their state descriptions). We follow Lee's state description in our discussion. He proposed a $(k+1)-$tuple state description of the form $(a_k, a_{k-1}, ..., a_1, a_q)$ where a_i indicates the number of servers with i stages remaining (i.e., a_k servers have k stages to complete, a_{k-1} servers have $k-1$ stages to complete and so on), for $1 \leq i \leq k$, and a_q indicates the number of customers in the queue waiting for service. We shall refer to this as Description 1.

Using Description 1, for a system with n servers and no customers waiting for service, i.e., $a_q = 0$, the number m of customers in the system (all of them being served) is given by

$$m = \sum_{i=1}^{k} a_i$$

where $m \leq n$. The number of idle servers is given by $n - m$. If $a_q > 0$, the number of customers in the system (either being served or waiting for service) is $m = n + a_q$, since

$$\sum_{i=1}^{k} a_i = n$$

and $m > n$. The total number of stages left in the system, l, can be obtained from the following formula

$$l = \sum_{i=1}^{k} a_i i + a_q k.$$

We now present an alternative state description, Description 2, that yields a more compact representation. Define the state by a three element descriptor (l, m, r), where l is the number of stages remaining from all customers in the system, m is the number of customers in the system and r is the pattern identifier needed when (l,m) is not enough to fully specify the state. The number of busy servers is given by the minimum of the number of customers in the system, m, and the number of servers, n. To better explain the situation of multiple patterns for a given combination of l and m, consider the following example:

Let the state of the system be $(11, 5, r)$ in an $M/E_3/5/5+q$ queue (see Figure 3-2). Thus, 11 uncompleted stages from 5 customers are distributed among the 5 available servers. This can occur in one of three ways, shown by the patterns (a), (b) and (c) of Figure 3-2. The number of shaded circles in each row in the figure denotes the number of uncompleted stages at a particular server. For example, pattern (a) has one server with three stages remaining to be completed and four servers with two stages remaining.

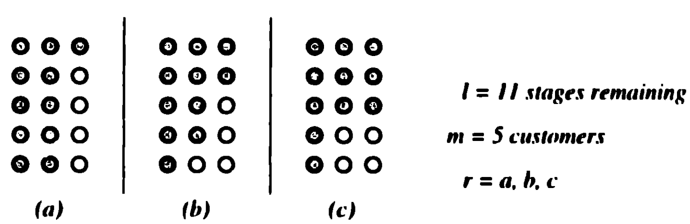


Figure 3-2: Alternative patterns for the state $(11, 5, r)$ in a $M/E_3/5$ Queue

In order to use Description 2, we need an algorithm to generate all possible patterns (and, thus, values of r) for the reduced state vector (l,m). Such an algorithm is provided below.

The two representations, Description 1 and Description 2, are equivalent. For example, state $(1,4,0,0)$ of Description 1 corresponds to state $(11,5,a)$ of Description 2, as shown in Figure 3-2. The disadvantage of Description 1 is that for larger Erlang orders, the $(k+1)-$tuple state description becomes long and complicated. An advantage of Description 1 is that the Chapman-Kolmogorov equations can be written quickly in a more systematic way than when Description 2 is used. The major disadvantage of Description 2 is that, as the Erlang order and the number of servers increase, the number of possible patterns for a particular (l,m) combination also increases rapidly. The advantage of Description 2, is that it leads to the derivation of the heuristics presented in Section 3.1.4. These derivations are shown in Sections 3.1.3 and 3.1.4.

Algorithm: Pattern Generator

This algorithm generates all patterns associated with specific values of l and m in Description 2. It works by assigning uncompleted stages to servers in a *left justified, top justified* manner such that each busy server has at least one uncompleted stage, and no servers have more than k stages.

To understand how the algorithm works, refer to Figure 3-3. Each pattern is a matrix of circles, with n rows and k columns, where the uncompleted stages are denoted by shaded circles as in figures 3-2 and 3-3. If there are fewer than n customers in the system, $m < n$ (thus, m busy servers), then the bottom $n - m$ rows contain only empty circles (the servers are idle). The rest of the servers will be assigned stages (the circles will be filled) according to the algorithm below.

The algorithm is as follows:

* Input data: l, m, n, k
* IF $m < n$
* THEN last $n - m$ rows are not used (idle servers)
* IF $m > n$
* THEN draw pattern for $l = l - k(m - n)$ uncompleted stages, FLAG ON

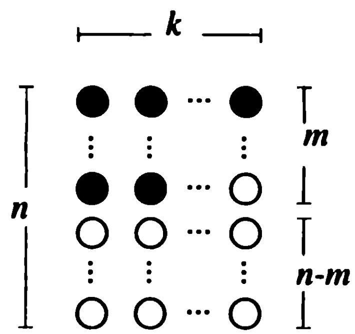


Figure 3-3: The $n \times k$ matrix of circles of a pattern when $m < n$

- stagesleft = l, column = k
- k^{th} COLUMN: minimum and maximum stages this pass
- smin[column] = minimum number of stages in column k
- smax[column] = maximum number of stages in column k
- stagesleft = stagesleft − smin[column]
- FOR stages[column] = smin[column] TO smax[column]
  - ⋄ column = column - 1
  - ⋄ $(k-1)^{st}$ COLUMN: minimum and maximum stages this pass
  - ⋄ smin[column] = minimum number of stages in column $k-1$
  - ⋄ smax[column] = maximum number of stages in column $k-1$
  - ⋄ stagesleft = stagesleft − smin[column]
  - ⋄ FOR stages[column] = smin[column] TO smax[column]
    - ∘ column = column - 1
    - ∘ $(k-2)^{nd}$ TO 2^{nd} COLUMNS: Nested loops as $(k-1)^{st}$ COLUMN loop
      - ⋆ 1^{st} COLUMN: No loop here since all stages left should be the same as the number of busy servers
      - ⋆ stages[column] = stagesleft
      - ⋆ PRINT PATTERN LOOP
      - ⋆ column = column + 1
    - ∘ stagesleft = stagesleft + smax[column] + 1
    - ∘ column = column + 1
  - ⋄ stagesleft = stagesleft + smax[column] + 1

◇ column = column + 1

* stagesleft = stagesleft + smax[column] + 1

* IF $m > n$

* THEN identify pattern obtained with original state (l,m), FLAG OFF

* END

The PRINT PATTERN LOOP has the following steps:

* FOR row = 1 TO $\min\{m, n\}$

  ◇ FOR col = 1 TO k

    ∘ IF stages[col] $\geq$ row

    ∘ THEN use location (row,col)

    ∘ ELSE do not use location (row,col)

* END

In the first loop of the algorithm, when $column = k$, the minimum and maximum number of uncompleted stages is given by

$$\begin{aligned} smin[column] &= \max\{0, stagesleft - (column - 1)p\} \\ smax[column] &= \min\left\{p, \left\lfloor \frac{stagesleft - p}{column - 1} \right\rfloor\right\}, \end{aligned}$$

where $p = min\{m, n\}$ indicates the number of busy servers. For the remaining $k - 2$ loops in the algorithm, for $2 \leq column \leq k - 1$, the minimum and maximum values are

$$\begin{aligned} smin[column] &= \max\{stages[column + 1], stagesleft - (column - 1)p\} \\ smax[column] &= \min\left\{p, \left\lfloor \frac{stagesleft - p}{column - 1} \right\rfloor\right\}. \end{aligned}$$

3.1.3 Exact Solution Technique

In this section, using Description 1, we derive the equations needed to obtain the state probabilities of the $M(t)/E_k(t)/n$ and $M(t)/E_k(t)/n/n + q$ systems. Let $\mathbf{S}_0$ be the array containing the state probabilities when $m < n$, and let $S_0(a_k, ..., a_1)$

be the probability of state $(a_k, ..., a_1, 0)$ for which $\sum_{i=1}^{k} a_i < n$. Let $\mathbf{Q}_0$ be the state probability array of states $(a_k, ..., a_1, 0)$ when $\sum_{i=1}^{k} a_i = n$, and $m = n$, with the state probabilities specified with $Q_0(a_k, ..., a_1)$.

Similarly, let $\mathbf{Q}_s$ be the state probability arrays for the case in which $a_q > 0$, and let $Q_s(a_k, ..., a_1)$ be the state probabilities of state $(a_k, ..., a_1, s)$ when there are s customers waiting for service in the queue. The total number of customers in the system in this case is $n + s$. If the system has infinite queue size, then $1 \leq s < \infty$. If the queue size is limited, $1 \leq s \leq q$, where q is the maximum number of customers that can wait for service.

For the case in which the queue size is limited, the array $\mathbf{Q}_q$ has the state probabilities $Q_q(a_k, ..., a_1)$ when there are $n+q$ customers in the system. We have specifically identified the arrays $\mathbf{Q}_0$, $\mathbf{Q}_s$ and $\mathbf{Q}_q$ because the transitions between states are different for the elements in each of these arrays.

Total Number of States

The arrays $\mathbf{S}_0$ and $\mathbf{Q}_s$ (for $0 \leq s \leq q$) are very sparse because most of their elements do not represent states in the queueing system. The total number of states in the limited queue size system that need to be considered is given by Equation 2.1, and is repeated below:

$$T_S = \binom{n+k}{n} + q \binom{n+k-1}{n}. \tag{3.1}$$

Note that the first term indicates the number of states when the queue is empty, and the second term indicates the number of states for the customers waiting in the queue. Table 3.1 shows various combinations of k, n and q, and their corresponding number of states which increase with the order of $max\{(n+q)!, q(n+k-1)!\}$. Notice that as the Erlang order, the number of servers and/or the size of the queue increase, the number of states in the system grows rapidly.

Table 3.1: Number of states for various k, n and q

k	n	q	Number of States
4	3	1	55
5	5	5	882
3	15	5	1,496
4	15	5	7,956
5	15	5	34,884
3	50	25	56,576
4	50	25	901,901

State-to-state Transitions

A transition between states occurs due to a stage completion or an arrival of a new customer to the system. The stage completion rate is $k\mu(t)$ and the arrival rate is given by $\lambda(t)$. Table 3.2 shows the transitions for each type of state and their corresponding state-to-state transition rates.

As mentioned earlier, the multidimensional arrays $\mathbf{S}_0$, $\mathbf{Q}_0$, $\mathbf{Q}_s$ and $\mathbf{Q}_q$ are very sparse. The number of elements in each k-dimensional array is $(n+1)^k$, for a total of $(q+2)(n+1)^k$ elements in all arrays, where only T_S elements (see Equation 3.1) are non-zero. The zero elements in the arrays represent the probabilities of the states that are impossible to reach due to the boundary conditions. Therefore, the a priori probabilities of such states are zero. As a result, the transitions to the zero probability states can be omitted in Table 3.2. The states with zero probability are described as follows.

Array S_0 represents states in which there are less than n customers in the system, $m < n$, and only m servers are busy. Therefore, in array S_0, the states with index sum greater than $n-1$, $\sum_{i=1}^{k} a_i > n-1$, and $a_q \neq 0$ have zero probability since there are only 0 to $n-1$ customers in service (and in the system) and no customers in the queue. On the other hand, arrays Q_s, for $0 \leq s \leq q$, represent the states in which there are $n+s$ customers in the system and all n servers are busy. Thus, the elements in Q_s representing states with index sum different from n, $\sum_{i=1}^{k} a_i \neq n$, and $a_q \neq s$ must equal zero and occur with zero probability. Therefore, transitions to and from

Table 3.2: State-to-State Transitions for the Exact Solution Technique

From State	In	To State	In	With Rate
$(a_k, ..., a_1, 0)$	$\mathbf{S}_0$	$(a_k + 1, a_{k-1}, ..., a_1, 0)$	$\mathbf{S}_0$	$\lambda(t)$ if $m < n - 1$
		$(a_k + 1, a_{k-1}, ..., a_1, 0)$	$\mathbf{Q}_0$	$\lambda(t)$ if $m = n - 1$
		$(a_k - 1, a_{k-1} + 1, a_{k-2}, ..., a_1, 0)$	$\mathbf{S}_0$	$a_k k\mu(t)$
		$(a_k, a_{k-1} - 1, a_{k-2} + 1, ..., a_1, 0)$	$\mathbf{S}_0$	$a_{k-1} k\mu(t)$
		$(a_k, ..., a_i - 1, a_{i-1} + 1, ..., a_1, 0)$	$\mathbf{S}_0$	$a_i k\mu(t)$ for $i = k...2$
		$(a_k, a_{k-1}, ..., a_1 - 1, 0)$	$\mathbf{S}_0$	$a_1 k\mu(t)$
$(a_k, ..., a_1, 0)$	$\mathbf{Q}_0$	$(a_k, ..., a_1, 1)$	$\mathbf{Q}_1$	$\lambda(t)$
		$(a_k, ..., a_i - 1, a_{i-1} + 1, ..., a_1, 0)$	$\mathbf{Q}_0$	$a_i k\mu(t)$ for $i = k...2$
		$(a_k, a_{k-1}, ..., a_1 - 1, 0)$	$\mathbf{S}_0$	$a_1 k\mu(t)$
$(a_k, ..., a_1, s)$	$\mathbf{Q}_s$	$(a_k, ..., a_1, s + 1)$	$\mathbf{Q}_{s+1}$	$\lambda(t)$
$(1 \leq s < q)$		$(a_k, ..., a_i - 1, a_{i-1} + 1, ..., a_1, s)$	$\mathbf{Q}_s$	$a_i k\mu(t)$ for $i = k...2$
		$(a_k + 1, a_{k-1}, ..., a_1 - 1, s - 1)$	$\mathbf{Q}_{s-1}$	$a_1 k\mu(t)$
$(a_k, ..., a_1, q)$	$\mathbf{Q}_q$	$(a_k, ..., a_i - 1, a_{i-1} + 1, ..., a_1, q)$	$\mathbf{Q}_q$	$a_i k\mu(t)$ for $i = k...2$
$(q < \infty)$		$(a_k + 1, a_{k-1}, ..., a_1 - 1, q - 1)$	$\mathbf{Q}_{q-1}$	$a_1 k\mu(t)$

states with zero probability do not occur. For example, the probability of entering a state in $\mathbf{Q}_0$ with index sum $\sum_{i=1}^{k} a_i = n - 1$ is exactly zero because the number of customers in the system is defined to be n and, therefore, the number of servers in use must be exactly n and not $n - 1$.

Chapman-Kolmogorov Equations

The state transitions specified in Table 3.2 lead directly to the Chapman-Kolmogorov equations for the $M(t)/E_k(t)/n$ and $M(t)/E_k(t)/n/n+q$ systems. The equations for solving the state probabilities are divided into five cases. First, when $\sum_{i=1}^{k} a_i = m < n - 1$, the equations are

$$\begin{aligned}
\dot{S}_0(a_k, ..., a_1)(t) \;=\; & -(\lambda(t) + mk\mu(t))S_0(a_k, ..., a_1)(t) \\
& +\lambda(t)S_0(a_k - 1, ..., a_1)(t) \\
& +\sum_{i=k}^{2}(a_i + 1)k\mu(t)S_0(a_k, ..., a_i + 1, a_{i-1} - 1, ..., a_1)(t) \\
& +(a_k + 1)k\mu(t)S_0(a_k, a_{k-1}, ..., a_1 + 1)(t).
\end{aligned} \tag{3.2}$$

Second, when $\sum_{i=1}^{k} a_i = m = n - 1$, they are given by

$$\begin{aligned}
\dot{S}_0(a_k, ..., a_1)(t) \;=\; & -(\lambda(t) + mk\mu(t))S_0(a_k, ..., a_1)(t) \\
& +\lambda(t)S_0(a_k - 1, ..., a_1)(t) \\
& +\sum_{i=k}^{2}(a_i + 1)k\mu(t)S_0(a_k, ..., a_i + 1, a_{i-1} - 1, ..., a_1)(t) \\
& +(a_k + 1)k\mu(t)Q_0(a_k, a_{k-1}, ..., a_1 + 1)(t)
\end{aligned} \tag{3.3}$$

and

$$\dot{S}_0(a_k, ..., a_1)(t) = 0 \qquad \text{if } \sum_{i=1}^{k} a_i \neq m, \text{ where } 0 \leq m \leq n - 1. \tag{3.4}$$

Once the number of customers equals the number of servers, $m = n$, the equations are

$$\dot{Q}_0(a_k, ..., a_1)(t) \;=\; -(\lambda(t) + nk\mu(t))Q_0(a_k, ..., a_1)(t) \tag{3.5}$$

$$
\begin{aligned}
&+\lambda(t)S_0(a_k-1,...,a_1)(t)\\
&+\sum_{i=k}^{2}(a_i+1)k\mu(t)Q_0(a_k,...,a_i+1,a_{i-1}-1,...,a_1)(t)\\
&+(a_k+1)k\mu(t)Q_1(a_k,a_{k-1},...,a_1+1)(t)
\end{aligned}
$$

and

$$
\dot{Q}_0(a_k,...,a_1)(t)=0 \qquad \text{if } \sum_{i=1}^{k} a_i \neq n. \tag{3.6}
$$

The equations of the state probabilities when there are customers waiting for service, $m > n$, are given by

$$
\begin{aligned}
\dot{Q}_s(a_k,...,a_1)(t) &= -(\lambda(t)+nk\mu(t))Q_s(a_k,...,a_1)(t) \\
&\quad+\lambda(t)Q_{s-1}(a_k,...,a_1)(t)\\
&\quad+\sum_{i=k}^{2}(a_i+1)k\mu(t)Q_s(a_k,...,a_i+1,a_{i-1}-1,...,a_1)(t)\\
&\quad+(a_k+1)k\mu(t)Q_{s+1}(a_k-1,a_{k-1},...,a_1+1)(t)
\end{aligned} \tag{3.7}
$$

and

$$
\dot{Q}_s(a_k,...,a_1)(t)=0 \qquad \text{if } \sum_{i=1}^{k} a_i \neq n, \tag{3.8}
$$

where $1 \leq s < q$, with $q = \infty$ if the queue size is unlimited. Finally, if $q < \infty$, then the following set of equations are needed when the number of customers in the system is $m = n + q$:

$$
\begin{aligned}
\dot{Q}_q(a_k,...,a_1)(t) &= -nk\mu(t)Q_q(a_k,...,a_1)(t) \\
&\quad+\lambda(t)Q_{q-1}(a_k,...,a_1)(t)\\
&\quad+\sum_{i=k}^{2}(a_i+1)k\mu(t)Q_q(a_k,...,a_i+1,a_{i-1}-1,...,a_1)(t)
\end{aligned} \tag{3.9}
$$

and

$$
\dot{Q}_q(a_k,...,a_1)(t)=0 \qquad \text{if } \sum_{i=1}^{k} a_i \neq n. \tag{3.10}
$$

Even though the Chapman-Kolmogorov equations are relatively straightforward to write from the state-to-state transitions, the number of equations to solve can be extremely large.

A detailed example of the $M/E_4/3/4$ queueing system is presented in Appendix A. In this Appendix, we show the state transition diagram and derive the state-to-state transitions with their corresponding probabilities. We strongly recommend that the reader consult Appendix A before proceeding to the following section in order to appreciate the complexity of the state transitions even for such a small system as the $M/E_4/3/4$ queue.

In Section 4.1, numerous examples of numerical solutions of Equations 3.2 through 3.10 are presented. The software used to solve numerically the state probabilities is described in Section 5.1.2.

3.1.4 Heuristic Solution Techniques

As seen in the previous section and in Appendix A, solving the $M/E_k/n/n+q$ queueing system may involve a large number of equations and very complicated state-to-state transitions. Such a large system of equations may require a long computational run to obtain a solution and, in some cases, it may even be too large to solve using currently available software and hardware. Because of this, we have developed two heuristics to reduce the number of simultaneous differential equations to solve. As a result, we are able to accurately solve large systems much faster, allowing solution of systems with numerous independent queues or even networks of queues (interconnected queues).

The two heuristics developed are described below. Both heuristics use Description 2, with states of the form (l, m, r), where l is the total number of stages remaining in the system, m is the number of customers in the system and r is the pattern identifier in case (l, m) is not unique.

Combination of States into a Single State

The basic idea in the heuristics is to reduce the number of equations by combining each collection of states (l, m, r) into a single state (l, m). This means that we shall not specify a unique state for every different pattern with the same (l, m).

When combining the multiple states into a single one, we modify also the transitions among states. Two possible transitions can occur when a customer completes a stage. If the stage completed was the last one needed for a customer to exit the service facility, the customer leaves the system and the system moves to a state with one customer less and one stage less to complete. The second type of transition is when a customer completes one stage of service but the customer remains in service. In this case, the system changes to a state with one stage less but with the same number of customers. Let the rates at which these transitions occur be $\alpha_{l,m}k\mu(t)$ and $\beta_{l,m}k\mu(t)$, respectively, where the subscript l, m indicates the state from which the transition originates.

The algorithm utilized to make the transformation from states (l, m, r) to state (l, m) is what differentiates the two heuristics, and the difference resides in the assumptions made to obtain the transition probabilities $\alpha_{l,m}$ and $\beta_{l,m}$. Let $\mathbf{P}$ be the array containing the state probabilities, and let $P_{l,m}$ be the state probability of state (l, m). As in the exact solution technique, the state probability array $\mathbf{P}$ is also very sparse.

Heuristic 1: Equally Likely Patterns (ELP)

The primary assumption in this heuristic is that all patterns, r, in states (l, m, r) are equally likely. Under this assumption, the state transition probabilities are as follows:

Suppose the system is in state (l, m). Let $\mathcal{D}$ be the set of all pattern identifiers that satisfy l and m, and let d be the total number of identifiers; let $s_{1,i}$ be the number of servers with only one stage remaining in pattern i; and let $\alpha_{l,m}$ be the transition probability to a state with $l-1$ stages and $m-1$ customers. (In the example of Figure 3-2, $\mathcal{D} = \{a, b, c\}$, $s_{1,a} = 0$, $s_{1,b} = 1$ and $s_{1,c} = 2$.) Hence,

$$\alpha_{l,m} = \frac{1}{pd} \sum_{i \in \mathcal{D}} s_{1,i}, \tag{3.11}$$

where $p = min\{m, n\}$ indicates the number of busy servers, is the transition probability of moving from state (l,m) to state $(l - 1, m - 1)$.

The transition probability from state (l,m) to state $(l - 1, m)$ is then given by

$$\beta_{l,m} = 1 - \alpha_{l,m}, \tag{3.12}$$

where both states have the same number of customers. The procedure for deriving the transition probabilities $\alpha_{l,m}$ and $\beta_{l,m}$, under Heuristic 1, is presented with an example in Appendix A.

Heuristic 2: Equally Likely Combinations (ELC)

For each pattern in state (l,m), we can count the number of different combinations of stages remaining in the servers. The number of combinations in pattern i is given by

$$C_i = \frac{p!}{p_1!p_2!...p_x!}, \tag{3.13}$$

where $p = min\{m,n\}$ is the number of busy servers, x is the number of different combinations of stages in the servers, and p_j denotes the number of servers with equal number of uncompleted stages, and $j = 1, 2, ..., x$. The total number of combinations for a particular state is given by

$$C_{total} = \sum_{i \in \mathcal{D}} C_i. \tag{3.14}$$

(In the example of Figure 3-2, in pattern (a), $x = 2$ and $C_a = 5$; in pattern (b), $x = 3$ and $C_b = 30$; and in pattern (c), $x = 2$ and $C_c = 10$.)

The fundamental assumption in this heuristic is that all the possible combinations of uncompleted stages in the servers, C_{total}, are equally likely. Under this assumption, the transition probability to a state with one fewer customer is then given by

$$\alpha_{l,m} = \frac{1}{pC_{total}} \sum_{i \in \mathcal{D}} s_{1,i} C_i \tag{3.15}$$

and $\beta_{l,m}$ is again as defined in Equation (3.12). An example deriving the transition probabilities $\alpha_{l,m}$ and $\beta_{l,m}$, using Heuristic 2, is also presented in Appendix A.

Total Number of States

If the system has unlimited queue space, there are an infinite number of states. To avoid an unstable queue, in this case we need to satisfy the relationship

$$\rho = \frac{\lambda(t)}{n\mu(t)} < 1.$$

For either heuristic, the reduced number of states and their transitions can be shown in a diagram. Figure 3-4 shows the state-transition diagram for the $M(t)/E_k(t)/n/n+q$ queue. For ease of reading, the time dependence has been omitted. In the top rows of the figure, we show the values of the number of customers, m; the values for the number of uncompleted stages, l, are shown inside the ovals.

Every customer that enters the system provides k stages to be completed before he/she leaves the service facility (see Figure 3-4). Therefore, we need to keep track of the number of servers in use to determine the service rate at which the system operates. We must separate the analysis into two sections: when $m \leq n$, the "*growing*" section, and when $m > n$, the "*queueing*" section. Note the *growing* and *queueing* sections in Figure 3-4. In this *growing* section the number of states increases each time a new customer arrives at the system and enters one of the available servers. When the $(n+1)^{th}$ customer arrives at the system, the *queueing* section starts. All servers (n total) remain busy when n or more customers are present in the system.

In the former case, the number of states with exactly m customers in the system and m servers occupied, is given by $m(k-1)+1$, and the total number of states in the *growing* section (up to n customers present) is

$$S_G = \sum_{m=0}^{n} [m(k-1)+1] = (k-1)\frac{n(n+1)}{2} + n + 1. \tag{3.16}$$

Every space for a customer waiting in queue provides $n(k-1)+1$ states to the *queueing* section and to the system.

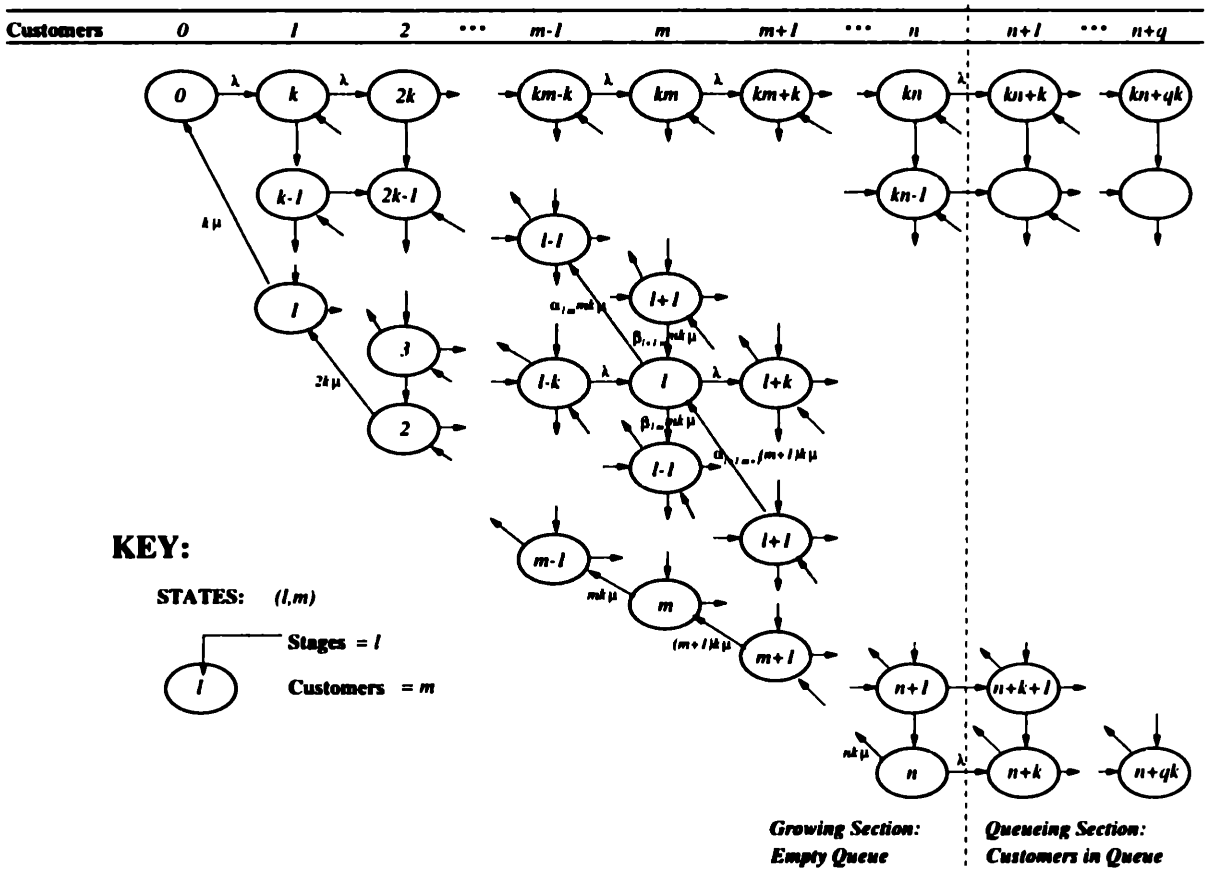


Figure 3-4: $M(t)/E_k(t)/n$ Queue

With $q < \infty$, the number of states in the *queueing* section is

$$S_Q = q[n(k-1)+1], \tag{3.17}$$

and the total number of states in system is represented by

$$\begin{aligned} S_S &= S_G + S_Q \\ S_S &= (k-1)\left[\frac{n(n+1)}{2} + qn\right] + q + n + 1. \end{aligned} \tag{3.18}$$

Notice that the number of states is of the order $max\{kn^2, knq\}$, generally much smaller than the number of states in the exact solution. As in the exact solution technique, this number depends on the number of servers, the Erlang order and the size of the queue.

Table 3.3: State-to-State Transitions for the Heuristic Solution Techniques

From State	To State	With Rate
(l,m)	$(l+k,m+1)$	$\lambda(t)$
	$(l-1,m-1)$	$\alpha_{l,m}mk\mu(t)$
	$(l-1,m)$	$\beta_{l,m}mk\mu(t)$
$(l-k,m-1)$	(l,m)	$\lambda(t)$
$(l+1,m+1)$		$\alpha_{l,m}mk\mu(t)$
$(l+1,m)$		$\beta_{l,m}mk\mu(t)$

State-to-State Transitions

In both heuristics, we need only to evaluate the state-to-state transition probabilities $\alpha_{l,m}$ and $\beta_{l,m}$ for the states in the *growing* section. This section of states includes the column of states where the number of customers is equal to the number of servers in the system. Each column of states in the *queueing* section has the same state-to-state transition probabilities $\alpha_{l,m}$ and $\beta_{l,m}$ as the column that has exactly n customers and n servers occupied. This is because the distribution of stages remaining in the queue (waiting to enter a server) does not affect the stage distribution for customers in service. The only difference occurs in the $(n+q)^{th}$ column where transitions to and from the right do not occur since no more than $n+q$ customers are allowed in the system.

As in the exact solution technique, state transitions occur when a customer arrives to the system (rate $\lambda(t)$) or when a service stage is completed (rate $k\mu(t)$). Table 3.3 summarizes the state-to-state transitions for the reduced number of states.

The total number of elements in the three-dimensional state probability array $\mathbf{P}$ is

$kn(n+q)^2$, and only S_S elements (see Equation 3.18) are non-zero. The zero elements represent the probabilities of the states that cannot be reached due to the boundary conditions, thus, the a priori probabilities of those states are zero. Therefore, the transitions described in Table 3.3 are not present in all states since the transitions to the zero probability states can be omitted. The states with zero probability are described as follows.

The states with $l > km$ when only m customers are present have probability zero since it is not possible to have more than k stages per customer in the system. Similarly, states with $l < m$, if $m \leq n$, or with $l < n + ik$, if there are i customers in the queue, have probability zero because it is not possible to have less than one uncompleted stage per customer in service and less than k uncompleted stages per customer in the queue. Hence, transitions to and from states with zero probability do not occur. Figure 3-4 shows all the states that do not have zero a priori probability. For example, in Figure 3-4, in the column with 2 customers, we cannot enter a state with only one uncompleted stage, $l = 1$, at the bottom of the column since we would have only one uncompleted stage for two customers. Note, as well, that a transition from a state with $l = 2k + 1$ to a state with $l = 2k$, with 2 customers, at the top of the column, cannot occur since we would have more than $2k$ uncompleted stages for only two customers. Therefore, the states $(1, 2)$ and $(2k + 1, 2)$ from the examples above, are not present in Figure 3-4.

Chapman-Kolmogorov Equations

Now, we can write the Chapman-Kolmogorov equations for both queues $M(t)/E_k(t)/n$ and $M(t)/E_k(t)/n/n+q$. Equations 3.19 through 3.25 represent the dynamics of the *growing* section. We first show the equations when the number of customers m is less than the number of servers in the system, $m < n$:

$$\dot{P}_{0,0}(t) = -\lambda(t)P_{0,0}(t) + k\mu(t)P_{1,1}(t) \tag{3.19}$$

$$\begin{aligned}\dot{P}_{mk,m}(t) &= -(\lambda(t)+mk\mu(t))P_{mk,m}(t) \\ &\quad +\lambda(t)P_{mk-k,m-1}(t) \\ &\quad +\alpha_{mk+1,m+1}(m+1)k\mu(t)P_{mk+1,m+1}(t)\end{aligned} \tag{3.20}$$

for

$$m = 1,2,...,(n-1),$$

$$\begin{aligned}\dot{P}_{mk-x,m}(t) &= -(\lambda(t)+mk\mu(t))P_{mk-x,m}(t) \\ &\quad +\lambda(t)P_{mk-k-x,m-1}(t) \\ &\quad +\beta_{mk-x+1,m}mk\mu(t)P_{mk-x+1,m}(t) \\ &\quad +\alpha_{mk-x+1,m+1}(m+1)k\mu(t)P_{mk-x+1,m+1}(t)\end{aligned} \tag{3.21}$$

for

$$\begin{aligned} m &= 1,2,...,(n-1), \\ x &= 1,2,...,(m-1)(k-1),\end{aligned}$$

$$\begin{aligned}\dot{P}_{mk-y,m}(t) &= -(\lambda(t)+mk\mu(t))P_{mk-y,m}(t) \\ &\quad +\beta_{mk-y+1,m}mk\mu(t)P_{mk-y+1,m}(t) \\ &\quad +\alpha_{mk-y+1,m+1}(m+1)k\mu(t)P_{mk-y+1,m+1}(t)\end{aligned} \tag{3.22}$$

for

$$\begin{aligned} m &= 1,2,...,(n-1), \\ y &= (m-1)(k-1)+1,(m-1)(k-1)+2,...,m(k-1),\end{aligned}$$

and m is the number of customers in the system.

When the number of customers equals the number of servers in the system, $m = n$, the equations are

$$\dot{P}_{nk,n}(t) = -(\lambda(t)+nk\mu(t))P_{nk,n}(t) \tag{3.23}$$

$$+\lambda(t)P_{(n-1)k,n-1}(t)$$
$$+\alpha_{(n-1)k+1,n}nk\mu(t)P_{nk+1,n+1}(t)$$

$$\begin{aligned}\dot{P}_{nk-x,n}(t) &= -(\lambda(t)+nk\mu(t))P_{nk-x,n}(t) \\ &\quad +\lambda(t)P_{(n-1)k,n-1}(t) \\ &\quad +\beta_{nk-x+1,n}nk\mu(t)P_{nk-x+1,n}(t) \\ &\quad +\alpha_{(n-1)k-x+1,n}nk\mu(t)P_{nk-x+1,n+1}(t)]\end{aligned} \tag{3.24}$$

where $x = 1, 2, ..., (n-1)(k-1),$

$$\begin{aligned}\dot{P}_{nk-y,n}(t) &= -(\lambda(t)+nk\mu(t))P_{nk-y,n}(t) \\ &\quad +\beta_{nk-y+1,n}nk\mu(t)P_{nk-y+1,n}(t)\end{aligned} \tag{3.25}$$

where $y = (n-1)(k-1)+1, (n-1)(k-1)+2, ..., n(k-1).$

Equations 3.26 through 3.28 capture the dynamics of the *queueing* section when the queue size is unlimited, $q = \infty$. The equations are given by

$$\begin{aligned}\dot{P}_{mk,m}(t) &= -(\lambda(t)+nk\mu(t))P_{mk,m}(t) \\ &\quad +\lambda(t)P_{mk-k,m-1}(t) \\ &\quad +\alpha_{(n-1)k+1,n}nk\mu(t)P_{mk+1,m+1}(t)\end{aligned} \tag{3.26}$$

for $m = (n+1), (n+2), ..., (n+q-1),$

$$\begin{aligned}\dot{P}_{mk-x,m}(t) &= -(\lambda(t)+nk\mu(t))P_{mk-x,m}(t) \\ &\quad +\lambda(t)P_{mk-k-x,m-1}(t)\end{aligned} \tag{3.27}$$

$$+\beta_{nk-x+1,n} nk\mu(t) P_{mk-x+1,m}(t)$$
$$+\alpha_{(n-1)k-x+1,n} nk\mu(t) P_{mk-x+1,m+1}(t)$$

for

$$\begin{aligned} m &= (n+1),(n+2),...,(n+q-1), \\ x &= 1,2,...,(n-1)(k-1), \end{aligned}$$

$$\begin{aligned} \dot{P}_{mk-y,m}(t) &= -(\lambda(t)+nk\mu(t))P_{mk-y,m}(t) \\ &\quad +\lambda(t)P_{mk-k-y,m-1}(t) \\ &\quad +\beta_{nk-y+1,n} nk\mu(t) P_{mk-y+1,m}(t) \end{aligned} \tag{3.28}$$

for

$$\begin{aligned} m &= (n+1),(n+2),...,(n+q-1), \\ y &= (n-1)(k-1)+1,(n-1)(k-1)+2,...,n(k-1). \end{aligned}$$

In the case with limited size queue, $q < \infty$, Equations 3.29 to 3.30 are needed. The equations for the states when there are $n+q$ customers in the system are:

$$\begin{aligned} \dot{P}_{(n+q)k,n+q}(t) &= -nk\mu(t)P_{(n+q)k,n+q}(t) \\ &\quad +\lambda(t)P_{(n+q-1)k,n+q-1}(t) \end{aligned} \tag{3.29}$$

$$\begin{aligned} \dot{P}_{(n+q)k-x,n+q}(t) &= -nk\mu(t)P_{(n+q)k-x,n+q}(t) \\ &\quad +\lambda(t)P_{(n+q-1)k-x,n+q-1}(t) \\ &\quad +\beta_{nk-x+1,n} nk\mu(t) P_{(n+q)k-x+1,n+q}(t) \end{aligned} \tag{3.30}$$

where

$$x = 1,2,...,n(k-1).$$

And finally,

$$\dot{P}_{l,m}(t) = 0 \tag{3.31}$$

for values (l, m) not included in Equations 3.19 to 3.30.

In Section 4.1, numerous examples with solutions of the above equations are presented and compared to solutions of Equations 3.2 to 3.10 of the exact model. The software used to solve numerically the state probabilities is described in Section 5.1.2. Appendix A illustrates graphically the mechanics involved in applying both the exact and heuristic solution techniques to the $M/E_4/3/4$ queueing system. Included are figures and explanations of all possible transitions and computations of associated α's and β's. The reader is urged to review this appendix.

3.2 Performance Measures of Interest

After solving the Chapman-Kolmogorov equations and obtaining the state transition probabilities for all states $(a_k, ..., a_1, a_q)$, of the exact solution technique, and all states (l, m), of the heuristic solution techniques, the results can be processed to calculate the probability distribution for the number of customers m in the system. The occupancy probabilities for the exact solution technique are given by

$$P_m(t) = \begin{cases} \sum_{\forall a_i \in \mathcal{M}} S_0(a_k, ..., a_1)(t) & 0 \leq m < n \\ \sum_{\forall a_i \in \mathcal{M}} Q_m(a_k, ..., a_1)(t) & n \leq m \leq q \end{cases} \tag{3.32}$$

where $\mathcal{M}$ is the set of indices a_i that satisfy $\sum_{i=1}^{k} a_i = m$. In the case of the heuristic solution technique, the occupancy probabilities are calculated as follows:

$$P_m(t) = \begin{cases} \sum_{l=m}^{km} P_{l,m}(t) & 0 \leq m \leq n \\ \sum_{l=n+(m-n)k}^{km} P_{l,m}(t) & n < m \leq q \end{cases} \tag{3.33}$$

The occupancy probabilities are used to obtain performance measures of interest that are classified into two categories: aggregate probabilities, and expected values. All the performance measures described below, with the exception of those based on Little's Formula, are valid for steady-state as well as transient conditions with either stationary or non-stationary parameters.

3.2.1 Aggregate Probabilities

Besides the probability distribution of customers in the system, we are interested in the following aggregate probabilities:

1. Probability of entering the queue: This is defined as the probability of a customer arriving to a finite capacity system when there are between n and $n+q-1$ customers present in the system. If there are already $n+q$ customers in the system, the arriving customer is rejected (or diverted) and does not enter the queue. The probability is obtained by

$$P(Queueing)(t) = \sum_{m=n}^{n+q-1} P_m(t).$$

2. Probability of a saturated queue: This is defined as the probability of having exactly $n+q$ customers in a finite capacity system when a new customer arrives. No more customers can enter the queue. The probability is given by:

$$P(Saturated)(t) = P_{n+q}(t).$$

These two probabilities will be used in the applications in Chapter 5.

3.2.2 Expected Values

Using the occupancy probabilities of Section 3.2, we can also calculate the expected number of busy servers at time t. This quantity is important to determine the workload of the system, and is given by

$$E[Busy](t) = \sum_{m=0}^{n} mP_m(t) + n\left(\sum_{m=n+1}^{n+q} P_m(t)\right) \qquad (3.34)$$

From the point of view of an observer of the system, the expected instantaneous delay of a customer that enters the queue at time t is given by

$$E[Delay](t) = \frac{1}{n\mu(t)} \left(\sum_{i=0}^{q-1} (i+1) P_{n+i}(t) \right). \tag{3.35}$$

We shall refer to this expected delay as *expected virtual delay.* In this formula, we only consider the customers that enter the queue because the amount of delay for the diverted or rejected customers is not known. This means that a customer entering the system where there are already $n+i$ customers, $0 \leq i \leq q-1$, has an expected virtual delay of $\frac{1}{n\mu(t)}(i+1)$ units of time.

We are also interested in the following quantities under constant demand and constant service rate:

- L = steady-state expected number of customers in the system;
- L_q = steady-state expected number of customers in the queue;
- W = steady-state expected time in the system, including service time; and,
- W_q = steady-state expected waiting time in the queue,

For an infinite capacity system, the system and queue statistics can be obtained using Little's Formula:

$$W = \frac{L}{\lambda}$$

and

$$W_q = \frac{L_q}{\lambda},$$

where

$$\begin{aligned} L &= \sum_{m=0}^{\infty} m P_m, \\ L_q &= \sum_{m=n}^{\infty} (m-n) P_m, \end{aligned}$$

λ is the arrival rate to the system, and P_m is the steady-state probability of having m of customers in the system.

If the queue has finite capacity,

$$L = \sum_{m=0}^{n+q} mP_m \tag{3.36}$$

and

$$L_q = \sum_{m=n}^{n+q} (m-n)P_m. \tag{3.37}$$

Since the queue size is limited, the effective arrival rate to the system is given by

$$\lambda' = \lambda(1 - P_{n+q}).$$

Then, using Little's Formula with the effective arrival rate λ', we obtain the expected waiting times in the system and in the queue, respectively, as follows:

$$\begin{aligned} W &= \frac{L}{\lambda'} \\ W_q &= \frac{L_q}{\lambda'}. \end{aligned}$$

The performance measures defined in this section are used in the validation of the heuristic solution techniques in Section 4.1, as well as in the application presented in Section 5.2.

3.3 Heuristic Solution for Systems with Variable Number of Servers

Most results in queueing theory applied to practical problems analyze systems with constant number of servers and assume that the capacity of the systems is time-invariant. A more realistic scenario is that service facilities experience fluctuations on the number of servers, and thus, causing the system capacity to increase or to decrease, due to variations in demand, server failures and system maintenance among many factors. For example, real-life problems in which we use queueing results and experience a variable number of servers could be a bank or an airline counter, where the number of tellers may vary during the day according to the expected demand (e.g., lunch time, airplane departures or arrivals); another example could be the number of tollbooths opened in a highway or the lines of traffic dedicated to one direction or the other, depending on the rush hour or the traffic characteristics; and air traffic operations (e.g., in airports or en-route sectors) where the number of active runways or the number of air traffic controllers depends on the time of the day, on the weather or even on the types of aircraft requesting service. This is the motivation for solving systems with variable number of servers.

Therefore, in this Section we examine a variant of the $M(t)/E_k(t)/n/n+q$ queueing system which includes a time-dependent number of servers: $n(t)$. We present two heuristic approaches: one each for the exact and ELP solution techniques presented in Section 3.1.

The following assumptions are required for both algorithms:

1. If a server is to be *closed*, it is equally likely to be any one of them: idle/busy status is not taken into account.

2. When a new server is *opened* and there are customers waiting for service, the first customer in queue enters the server.

3. Servers are closed or opened one at a time. The heuristic may be repeated iteratively, however, to obtain the desired number of servers opened or closed in any particular time period.

4. If the server closed is serving a customer, we ignore that customer from the instant the server is closed. The customer eventually leaves the system.

5. Changes in the number of servers do not occur frequently.

Assumptions 1 through 3 are self-explanatory and relate to the operation of the heuristics at the time the number of servers in the system changes. Assumption 4 is a simplifying assumption and deserves particular attention. If Assumption 4 is not used, we would need to consider the customer being served by the closed server when obtaining the distribution of customers in the system, as well as when evaluating any performance measure of interest. If Assumption 5 is in effect, the number of customers "ignored" due to changes in the number of servers is small and their statistics can be safely neglected. Therefore, Assumptions 4 and 5 must be used together. Assumption 5 is a realistic assumption because in most practical applications, the number of servers do not change often. In the long-run, after the transients disappear, the system behaves as if the original number of servers would have been always the same as the number of servers in the modified system.

Both heuristics map the original state probability arrays into corresponding new state probability arrays. The mapping distributes the state probabilities of the original system, at the time the number of servers changes, into the initial probabilities for the modified system. If a server is closed, the mapping is made so that the system moves from a state with l stages remaining and n servers to a state with $l - c$ stages remaining and $n - 1$ servers where $0 \leq c \leq k$ is the number of stages remaining in the closed server. On the other hand, if a server is opened, the mapping is made so that the system moves from a state with l stages remaining and n servers to a state with l stages remaining and $n + 1$ servers. The probabilities of change between states in the original and modified systems are described in the algorithms below.

Heuristic 3: Variable n in the Exact Solution Technique

This heuristic maps the state probabilities from the old to the new systems when using the exact solution technique. We start by defining the state probability arrays for both systems. Let $P_{states1}(a_k, ..., a_1, a_q)$ be the probability of state $(a_k, ..., a_1, a_q)$ in the original state probability arrays; let $P_{states2}(a_k, ..., a_1, a_q)$ be the probability of state $(a_k, ..., a_1, a_q)$ in the modified state probability arrays ; and let n_1 and n_2 be the number of servers in the original and modified systems, respectively. The algorithm followed by Heuristic 3 is given below:

* Input data: n_1, n_2, k, q, $P_{states1}(a_k, ..., a_1, a_q)$

* Initialize $P_{states2}(a_k, ..., a_1, a_q)$ to zero for all states $(a_k, ..., a_1, a_q)$

* IF $n_1 > n_2$ THEN

  ◇ FOR $c = 0$ TO $c = n_1 - 1$ DO

∘ Generate all states $(a_k, ..., a_1, 0)$ for which $\sum_{i=1}^{k} a_i = c$

∘ For each state:

⋆ FOR $i = 1$ TO $i = k$ DO

▷ IF $a_i > 0$ THEN

$P_{states2}(a_k, ..., a_i - 1, ..., a_1, 0) =$

$$\frac{a_i}{n_1} P_{states1}(a_k, ..., a_1, 0) + P_{states2}(a_k, ..., a_i - 1, ..., a_1, 0)$$

⋆ For the idle servers $(a_i = 0)$:

$P_{states2}(a_k, ..., a_1, 0) =$

$$\frac{n_1 - c}{n_1} P_{states1}(a_k, ..., a_1, 0) + P_{states2}(a_k, ..., a_1, 0)$$

⋄ FOR $c = n_1$ TO $c = n_1 + q$ DO

∘ Generate all states $(a_k, ..., a_1, a_q)$ for which $\sum_{i=1}^{k} a_i = n_1$ and $a_q = c - n_1$

∘ For each state:

⋆ FOR $i = 1$ TO $i = k$ DO

▷ IF $a_i > 0$ THEN

$P_{states2}(a_k, ..., a_i - 1, ..., a_1, a_q) =$

$$\frac{a_i}{n_1} P_{states1}(a_k, ..., a_1, a_q) + P_{states2}(a_k, ..., a_i - 1, ..., a_1, a_q)$$

∗ IF $n_1 < n_2$ THEN

⋄ FOR $c = 0$ TO $c = n_1 - 1$ DO

∘ Generate all states $(a_k, ..., a_1, 0)$ for which $\sum_{i=1}^{k} a_i = c$

∘ For each state:

$P_{states2}(a_k, ..., a_1, 0) = P_{states1}(a_k, ..., a_1, 0)$

⋄ FOR $c = n_1$ TO $c = n_1 + q$ DO

∘ Generate all states $(a_k, ..., a_1, a_q)$ for which $\sum_{i=1}^{k} a_i = n_1$ and $a_q = c - n_1$

∘ For each state:

$P_{states2}(a_k + 1, ..., a_1, a_q - 1) = P_{states1}(a_k, ..., a_1, a_q)$

∗ END

For example, if we have an $M(t)/E_3(t)/4/4+q$ queueing system and it changes to an $M(t)/E_3(t)/3/3+q$ queueing system, the probability of being in state $(1, 1, 2, 0)$ when the system is modified is divided into the new states $(0, 1, 2, 0)$, $(1, 0, 2, 0)$ and

$(1,1,1,0)$, with the ratios $\frac{1}{4}$, $\frac{1}{4}$ and $\frac{1}{2}$, respectively, given by the probabilities of changing to those states, as shown in Figure 3-5.

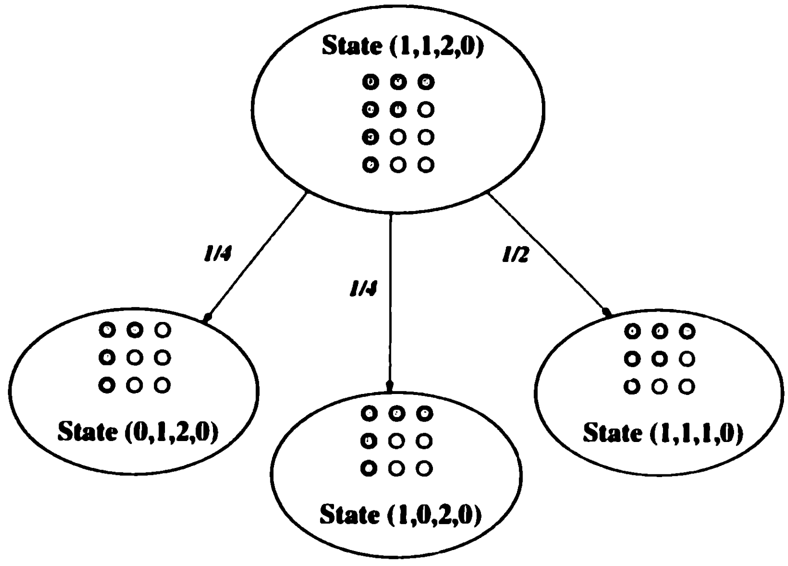


Figure 3-5: Mapping state $(1,1,2,0)$ in $M(t)/E_3(t)/4/4+q$ into states $(0,1,2,0)$, $(1,0,2,0)$ and $(1,1,1,0)$ in $M(t)/E_3(t)/3/3+q$

In the case that we modify an $M(t)/E_3(t)/4/4+q$ queueing system by increasing the number of servers, i.e., to an $M(t)/E_3(t)/5/5+q$ queueing system, the probability of being in state $(2,1,1,1)$ in the original system is assigned completely to state $(3,1,1,0)$ in the modified system.

Heuristic 4: n-Variable in the Heuristic Solution Techniques

We now introduce the heuristic to map the original and modified systems when the number of servers is changed. We define the state probabilities as follows. Let $P_{states1}(l,m)$ be the probability of state (l,m) in the original state probability arrays; let $P_{states2}(l,m)$ be the probability of state (l,m) in the modified state probability arrays ; and let n_1 and n_2 be the number of servers in the original and modified systems, respectively. Heuristic 4 has the following algorithm:

* Input data: n_1, n_2, k, q, $P_{states1}(l,m)$
* Initialize $P_{states2}(l,m)$ to zero for all states (l,m)
* Let $stages1 = stages2 = 0$

* IF $n_1 > n_2$ THEN
  - ◇ $P_{states2}(0,0) = P_{states1}(0,0)$
  - ◇ FOR $c = 1$ TO $c = n_1$ DO
    - ∘ $stages1 = c * k$
    - ∘ FOR $j = 0$ TO $j = c * (k-1)$ DO
      - ⋆ Generate patterns for state $(stages1 - j, c)$ with n_1 and k
      - ⋆ Obtain the probability for each pattern (P_{pat}) using ELP of ELC
      - ⋆ For each pattern:
        - ▷ FOR $i = 1$ TO $i = c$ DO
          - • Remove row i in pattern
          - • $stages2$ = stages left in remaining rows of pattern
          - • $c_2 = c - 1$
          - • $P_{states2}(stages2, c_2) =$

            $\frac{1}{n_1} P_{states1}(stages1, c) * P_{pat} + P_{states2}(stages2, c_2)$
        - ▷ IF $c_1 < n_1$ THEN

          $P_{states2}(stages1, c_1) =$

          $\frac{n_1 - c}{n_1} P_{states1}(stages1, c) * P_{pat} + P_{states2}(stages1, c)$
  - ◇ FOR $c = n_1 + 1$ TO $c = n_1 + q$ DO
    - ∘ $stages1 = c * k$
    - ∘ $w = c - n_1$
    - ∘ FOR $j = 0$ TO $j = n_1 * (k-1)$ DO
      - ⋆ Generate patterns for state $(stages1 - w * k - j, n_1)$ with n_1 and k
      - ⋆ Obtain the probability for each pattern (P_{pat}) using ELP of ELC
      - ⋆ For each pattern:
        - ▷ FOR $i = 1$ TO $i = n_1$ DO
          - • Remove row i in pattern
          - • $stages2$ = (stages left in remaining rows of pattern) + $w * k$
          - • $c_2 = c - 1$
          - • $P_{states2}(stages2, c_2) =$

            $\frac{1}{n_1} P_{states1}(stages1, c) * P_{pat} + P_{states2}(stages2, c_2)$

* IF $n_1 < n_2$ THEN
  ⋄ FOR $c = 1$ TO $c = n_1$ DO
    ∘ $stages1 = c * k$
    ∘ FOR $j = 0$ TO $j = c * (k - 1)$ DO
      ⋆ $P_{states2}(stages1 - j, c) = P_{states1}(stages1 - j, c)$
  ⋄ FOR $c = n_1 + 1$ TO $c = n_1 + q$ DO
    ∘ $stages1 = c * k$
    ∘ FOR $j = 0$ TO $j = n_1 * (k - 1)$ DO
      ⋆ $P_{states2}(stages1 - j, c) = P_{states1}(stages1 - j, c)$
* END

Let us use again the example used for Heuristic 3. If we are in state $(8, 4)$ in an $M(t)/E_3(t)/4/4 + q$ queueing system and the number of servers is reduced to 3, the probability of being in the state $(8, 4)$ must map into states $(7, 3)$, $(6, 3)$ and $(5, 3)$ in the $M(t)/E_3(t)/3/3 + q$ model, as shown in Figure 3-6. Note that state $(8, 4)$ has possible patterns (a), (b) and (c), in the original system. For example, pattern (b) maps to all three states in the new model in the following way: if the server with three remaining stages is closed, $\frac{1}{4}$ of the probability of being in the old state $(8, 4)$ is assigned to the new state $(5, 3)$; if either one of the two servers with two stages remaining is closed, $\frac{1}{2}$ of the probability of being in the old state $(8, 4)$ is assigned to the new state $(6, 3)$; if the server with one remaining stage is closed, $\frac{1}{4}$ of the probability of being in the old state $(8, 4)$ is assigned to the new state $(7, 3)$.

If the number of servers is increased by one, the states in the original system map into exactly the same state in the modified system. For example, the probability of state $(11, 5)$ in system $M(t)/E_3(t)/4/4 + q$ is assigned only to state $(11, 5)$ in system $M(t)/E_3(t)/5/5 + q$.

In general, if we have two systems with the same Erlang order k and the same queue size q, but with different number of servers, the set of states in the system with fewer servers is a subset of the set of states of the larger system. If the number of servers is decreased, the states in the larger system all map into the states in the

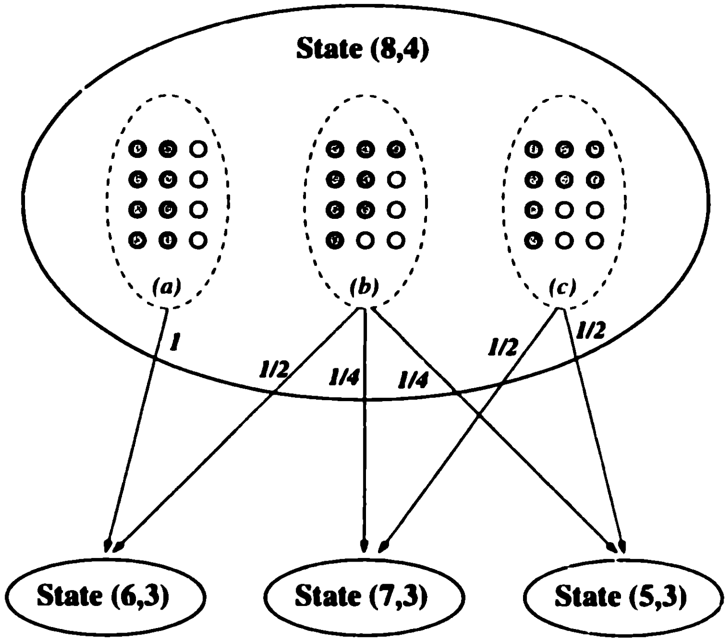


Figure 3-6: Mapping state $(8,4)$ in $M(t)/E_3(t)/4/4+q$ into states $(7,3)$, $(6,3)$ and $(5,3)$ in $M(t)/E_3(t)/3/3+q$

smaller system, i.e., every state probability in the smaller system is initialized with state probabilities of one or more states in the larger system. If the number of servers is increased, the states in the smaller system do not all map into the larger system. Therefore, those states that are not mapped from a smaller system are initialized with probability zero.

To understand this situation, consider the mapping of the $M(t)/E_3(t)/4/4+3$ system to the $M(t)/E_3(t)/5/5+3$. The system with $n=4$ has 80 states while the system with $n=5$ has 119 in the exact solution. For example, the probability of state $(0,0,5,0)$ in the $M(t)/E_3(t)/5/5+3$ system cannot be initialized with the state probabilities of the original system; instead, it is initialized with probability zero. Using the heuristic solution, the system with $n=4$ has 52 states and the one with $n=5$ has 69 states. In this case, the probability of state $(5,5)$ in the new system cannot be initialized with state probabilities in the smaller system and is initialized with zero probability.

When mapping a smaller system into a larger system using the heuristic solution, another interesting situation occurs. Even though a state may be initialized from the corresponding state in the smaller system, not all the patterns in the new state can be mapped from the patterns in the original state, but they are all initialized. For example, the state $(10, 5)$ in the $M(t)/E_3(t)/4/4 + 3$ queue has two patterns; the state $(10, 5)$ in the $M(t)/E_3(t)/5/5 + 3$ model has three different patterns. As seen in Figure 3-7, the two patterns in the original state map only into patterns (b) and (c) in the new one. Notice that the customer waiting for service in the queue in the original model has already entered a server in the modified system (patterns (b) and (c)). When we initialize state $(10, 5)$ in the $M(t)/E_3(t)/5/5 + 3$ model, we are initializing all patterns even though we have not mapped pattern (a). Because of the nature of the heuristic solution technique, we cannot differentiate the probabilities for each pattern individually. Therefore, we cannot choose to avoid initializing the unmapped patterns as the entire state is initialized as a whole.

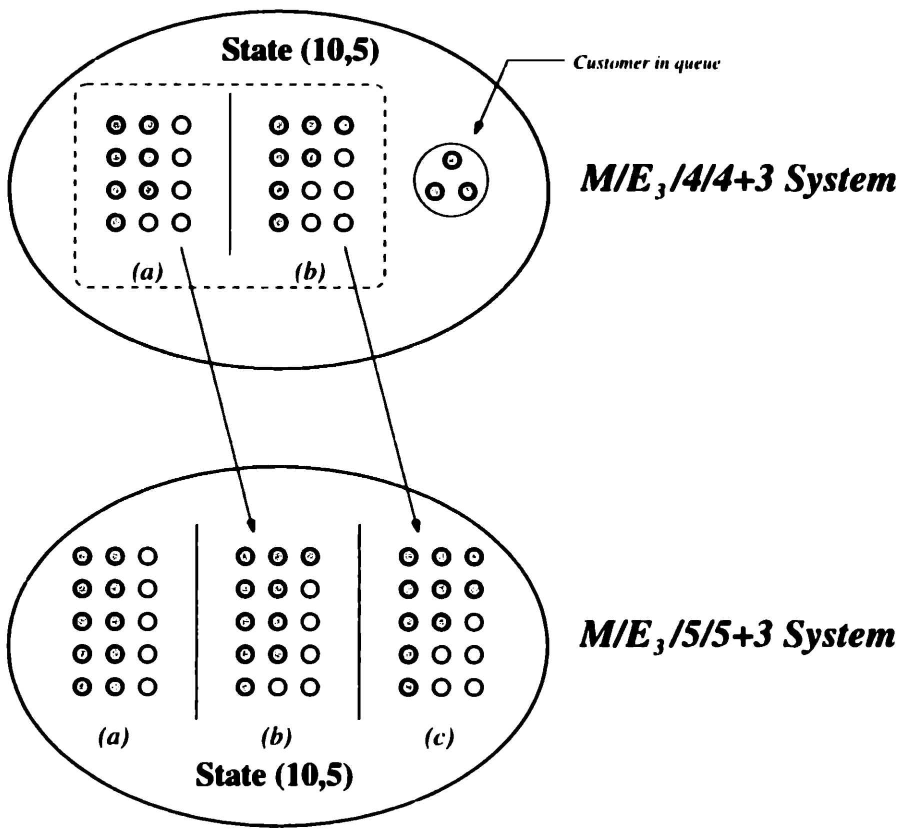


Figure 3-7: Initializing state $(10, 5)$

In Chapter 4, we implement both algorithms under stationary and nonstationary conditions. In the case of stationary parameters, we compare implementation results of both algorithms with the corresponding systems with a fixed number of servers, given that all systems have reached steady-state.

Chapter 4

Validation of the Heuristics

In this Chapter, we examine the performance of two heuristics introduced in Chapter 3. The primary validation consists of comparing steady-state numerical results obtained using the exact solution technique with those from Heuristics 1 and 2, under stationary conditions. An extensive collection of models, with varied parameters, is examined. We will show that Heuristic 2, Equally Likely Combinations (ELC), provides an excellent approximation to the exact results. As ELC requires the solution of fewer equations than the exact method, we will show that results are obtained much faster, and that larger models than currently solvable using the exact solution technique can be solved using ELC.

As a secondary study, we examine the accuracy of the transient behavior generated using the ELC solution technique for models with both stationary and nonstationary parameters. Finally, we examine performance of Heuristics 3 and 4 described at the end of Chapter 3 for systems with variable number of servers. We will see that ELC approximates well the exact results of the $M(t)/E_k(t)/n/n+q$ queueing system for all cases described above.

4.1 Validation: Stationary Conditions

We begin by describing the methodology used to compare the heuristic solution techniques to the exact solution technique. Using the equations developed in Chapter 3 for both the exact and heuristic solution techniques, we run an extensive set of experiments and compare directly the results of all three cases: Exact, Equally Likely Patterns (ELP) and Equally Likely Combinations (ELC).

Note that it is not possible to solve numerically an infinite number of Chapman-Kolmogorov equations. In order to obtain a finite number of Chapman-Kolmogorov equations we must limit the number of customers that enter the queue. This means that in all cases we solve the $M(t)/E_k(t)/n/n+q$ queue, $q < \infty$, where q is the maximum number of customers waiting for service. In practice, if q is sufficiently large and $\rho < 1$, the $M(t)/E_k(t)/n/n+q$ queue will provide an acceptable approximation to the $M(t)/E_k(t)/n$ queue. Included in the experiments are many examples with effectively infinite capacity.

A total of 510 experiments are presented here, which correspond to 170 different models solved using each solution technique (Exact, ELC and ELP). The models were selected to cover a wide range of system parameters. We include large and small systems in terms of both the number of servers, n, and the length of queue, q; large and small Erlang orders k to provide multiple distribution shapes for the service times; and a range of utilization ratios ρ including under- and over-saturated systems. As will be seen, the ELC solution technique performs extremely well for all parameters used.

We compare several measures of performance generated through use of these techniques. Using the occupancy probabilities defined in Section 3.2 we have calculated performance measures to extend the comparison to both aggregate probabilities and queue statistics.

Let us define an *epoch* as the basic unit of time used for observing model behavior. Both arrival and service rates are defined per unit of time and system performance is reported at the end of each epoch. For example, the service rate used in all models in this section is $\mu = 6$ per epoch. Therefore, we expect that a continuously busy server processes an average of 6 customers per epoch. All results presented here are for steady-state behavior. We assume that the models have effectively achieved steady-state when the state probabilities are constant, up to 6 decimal places, for at least 5 contiguous epochs. All systems start empty and idle.

Occupancy Probabilities

The following 11 examples illustrate that both heuristic solution techniques can provide accurate values of the steady-state occupancy probabilities. The first set of examples uses the $M/E_3/5/5 + 5$ queueing system. Figure 4-1 compares the three

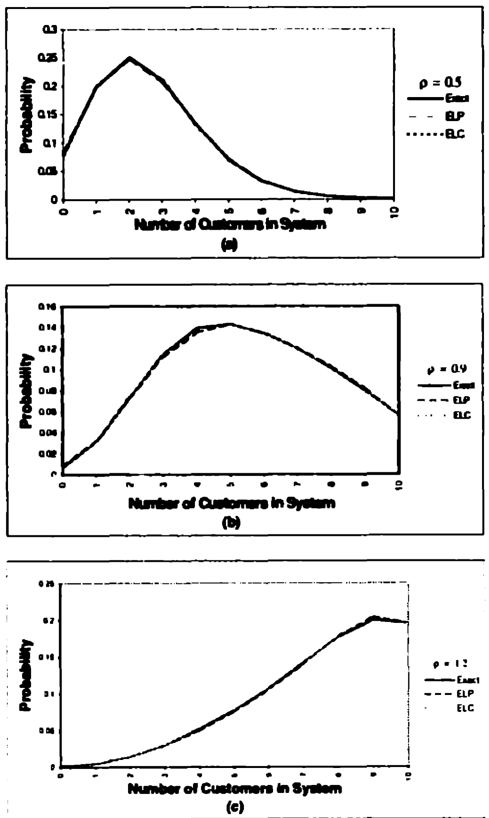


Figure 4-1: $M/E_3/5/5 + 5$ queueing system with (a) $\rho = 0.5$, (b) $\rho = 0.9$ and (c) $\rho = 1.2$

solution techniques (exact, ELP and ELC) with $\rho = 0.5$, 0.9 and 1.2. In all three plots in Figure 4-1, the only heuristic noticeable different from the exact curve is the ELP. These examples have a small Erlang order, a small number of servers and a very capacitated queue. The purpose of presenting this set of three examples is to illustrate the performance of the heuristic solution techniques in approximating the exact results for a small system, in terms of n and q, with various utilization ratios.

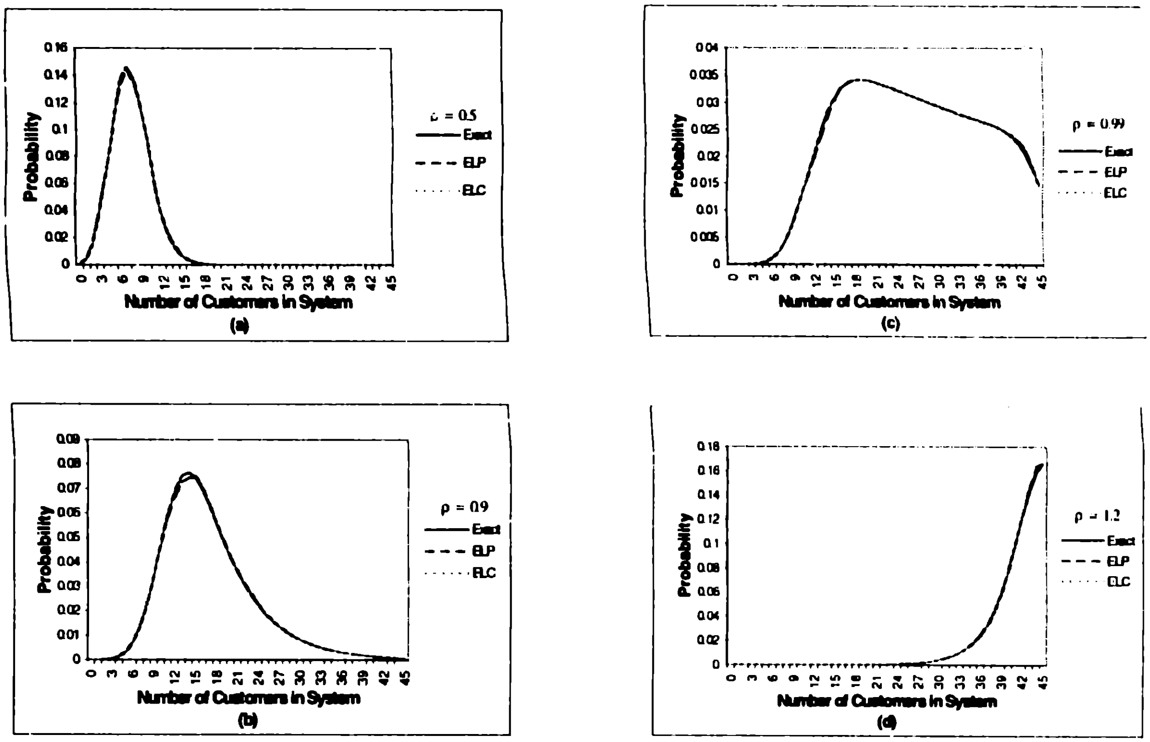


Figure 4-2: $M/E_4/15/15+30$ queueing system with $\rho = 0.5$, 0.9, 0.99 and 1.2

The second set of examples illustrates the system $M/E_4/15/15+30$. Figure 4-2 shows the results for four different utilization ratios, $\rho = 0.5$, 0.9, 0.99 and 1.2, using the exact, ELP and ELC approaches. In this Figure, parts (a) and (b), the curve of the heuristic technique ELP is the only one noticeable different from the exact values. Notice that in this set of examples with larger queue size and more servers in the system, the heuristics follow very closely the results obtained in the exact solution. Our next example, with the $M/E_5/3/3+100$ queue, is presented to compare the exact results with those obtained using the heuristic solution techniques for a system with a very large queue and a high utilization ratio, $\rho = 0.9$. Figure 4-3 illustrates the occupancy probabilities for this example.

The last example, the $M/E_{20}/3/3+10$ system, is presented in Figure 4-4. The purpose of presenting this example is to show that even for large k the steady-state occupancy probabilities are approximated well by the heuristic solution techniques. Figure 4-4 shows results for three utilization ratios, $\rho = 0.5$, 0.9 and 1.2. Note that both ELP and ELC curves are noticeably different than the exact curve in parts (b)

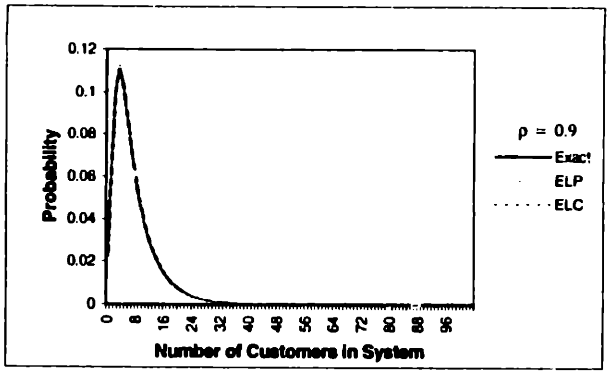


Figure 4-3: $M/E_5/3/3+100$ queueing system with $\rho = 0.9$

and (c) of Figure 4-4.

In all examples presented, the occupancy probabilities generated by the three solution techniques are extremely similar, showing that the heuristics may very well be used to solve the queueing models. Also from the examples above, we can see that even as the parameters are changed, the performance of the heuristics remains very good in approximating the exact results. There is no evidence that the performance of the heuristics will worsen for any particular change in the parameters.

The number of epochs required to reach steady-state in each of the 11 examples presented is given in Table 4.1. Notice that as the utilization ratio approaches 1, the number of epochs to reach steady-state increases, but when the queue is over-saturated, $\rho > 1$, the number of epochs is reduced; this may be because the system saturates faster and diverts traffic due to a limited queue size; thus, the probability of a saturated queue reaches its steady-state value in less time.

Aggregate Performance Measures

The Tables presented in this section illustrate the accuracy of both heuristic solution techniques for several additional performance measures. We have selected five aggregate performance measures to use in comparing the 170 models that provide the foundation for this validation study:

- Expected queue length, L_q
- Expected virtual delay, $E[Delay]$

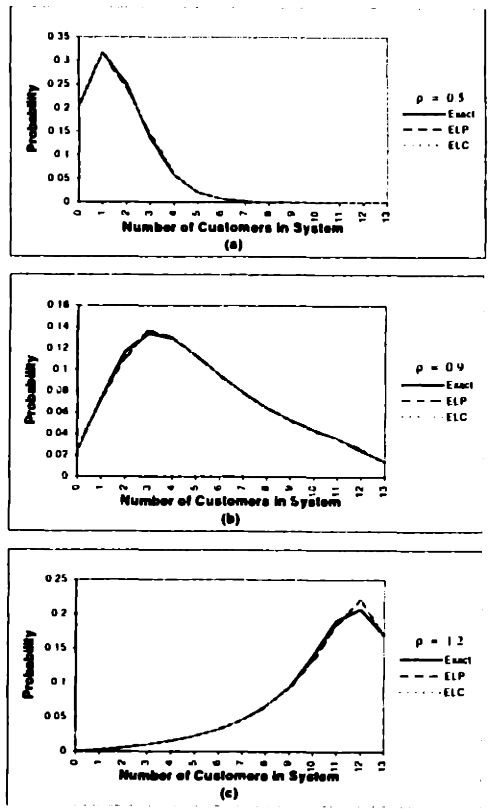


Figure 4-4: $M/E_{20}/3/3+10$ queueing system with (a) $\rho = 0.5$, (b) $\rho = 0.9$ and (c) $\rho = 1.2$

- Expected waiting time in the system, W
- Expected number of customers in the system, L
- Expected number of busy servers, $E[Busy]$

All of these statistics are defined in Section 3.2. Tables 4.2 through 4.36 show the results for the above statistics and are discussed below. All the Tables have the following format: columns 1 to 4 define the model for which the results are presented (k, n, q and ρ); columns 5 to 7 indicate the numerical results for the particular performance measure for the three solution techniques (exact, ELP and ELC); and, columns 8 and 9 indicates the percentage difference between the heuristic solutions and the exact solution, using the formula

$$\%\text{Difference} = 100 \times \frac{\text{Exact} - \text{Heuristic}}{\text{Exact}}$$

Table 4.1: Number of Epochs to Reach Steady-State

System	Utilization ratio ρ	Exact	ELC	ELP
$M/E_3/5/5+5$	0.5 0.9 1.2	3 4 4	3 4 4	3 4 4
$M/E_4/15/15+30$	0.5 0.9 0.99 1.2	2 12 24 9	2 12 24 9	2 12 24 9
$M/E_5/3/3+100$	0.9	100	99	99
$M/E_{20}/3/3+10$	0.5 0.9 1.2	3 17 15	3 17 15	3 16 15

where Exact is the value in Column 5 and Heuristic is replaced by the value of ELP or of ELC in column 6 or 7, respectively.

Values for the expected queue length are presented in Tables 4.2 to 4.8. Notice that using Heuristic 2, ELC, L_q is always within 3% of the exact solution. In most of the examples using Heuristic 1, ELP, L_q is within 4% of its exact value. Note, however, that there are sporadic cases in which the ELP error is as high as 16%. As can be seen in the Tables, even though both ELC and ELP generally provide accurate approximations of the exact model's behavior, ELC provides more consistent and, in most cases, more accurate results than ELP.

Tables 4.9 through 4.15 show results for the expected virtual delay. Similarly to the results for L_q, the ELC approach provides a better approximation yielding values always within 2% of the exact value. As before, most ELP values are good, within 5%, but there are some cases for which ELP values differ dramatically from the exact.

Table 4.2: Expected Queue Length, Part 1

k	n	q	ρ	Number of Customers Exact	Number of Customers ELP	Number of Customers ELC	% Difference ELP	% Difference ELC
3	3	1	0.50	0.0554	0.0562	0.0549	-1.50	0.85
3	3	1	0.90	0.2034	0.2039	0.2024	-0.21	0.49
3	3	1	1.20	0.3178	0.3176	0.3167	0.06	0.34
3	3	10	0.50	0.1669	0.1707	0.1662	-2.30	0.38
3	3	10	0.90	2.7443	2.7571	2.7441	-0.46	0.01
3	3	10	0.99	4.1709	4.1840	4.1728	-0.31	-0.05
3	3	10	1.20	6.9587	6.9714	6.9651	-0.18	-0.09
3	3	25	0.50	0.1670	0.1708	0.1663	-2.30	0.38
3	3	25	0.90	4.5286	4.5437	4.5263	-0.33	0.05
3	3	25	0.99	10.8896	10.9052	10.8911	-0.14	-0.01
3	3	25	1.20	21.4770	21.4902	21.4844	-0.06	-0.03
3	3	50	0.50	0.1670	0.1708	0.1663	-2.30	0.38
3	3	50	0.90	4.9247	4.9401	4.9218	-0.31	0.06
3	3	50	0.99	20.9389	20.9446	20.9287	-0.03	0.05
3	3	50	1.20	46.4603	46.4736	46.4677	-0.03	-0.02
3	3	100	0.50	0.1670	0.1708	0.1663	-2.30	0.38
3	3	100	0.90	4.9404	4.9558	4.9375	-0.31	0.06
3	3	100	0.99	36.6405	36.5621	36.5438	0.21	0.26
3	3	100	1.20	96.4603	96.4735	96.4676	-0.01	-0.01
3	5	5	0.50	0.0905	0.0940	0.0901	-3.89	0.45
3	5	5	0.90	1.2756	1.2899	1.2782	-1.12	-0.21
3	5	5	1.20	2.7011	2.7186	2.7096	-0.65	-0.32
3	5	15	0.50	0.0950	0.0986	0.0945	-3.85	0.52
3	5	15	0.90	3.3799	3.4008	3.3792	-0.62	0.02
3	5	15	0.99	6.2569	6.2798	6.2603	-0.37	-0.05
3	5	15	1.20	11.5533	11.5759	11.5639	-0.20	-0.09
3	5	50	0.50	0.0950	0.0986	0.0945	-3.85	0.52
3	5	50	0.90	4.6153	4.6395	4.6139	-0.52	0.03
3	5	50	0.99	20.6813	20.7102	20.6854	-0.14	-0.02
3	5	50	1.20	46.3898	46.4128	46.4009	-0.05	-0.02

Table 4.3: Expected Queue Length, Part 2

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
3	10	5	0.50	0.0262	0.0284	0.0261	-8.30	0.72
3	10	5	0.90	1.0374	1.0569	1.0412	-1.88	-0.37
3	10	5	1.00	1.5626	1.5851	1.5696	-1.44	-0.45
3	10	5	1.20	2.5646	2.5909	2.5767	-1.03	-0.47
3	15	5	0.50	0.0085	0.0096	0.0084	-13.11	0.94
3	15	5	0.90	0.8885	0.9088	0.8923	-2.29	-0.43
3	15	5	1.20	2.4832	2.5130	2.4965	-1.20	-0.54
3	15	30	0.50	0.0090	0.0102	0.0089	-13.11	1.00
3	15	30	0.90	3.5290	3.5720	3.5269	-1.22	0.06
3	15	30	0.99	12.0455	12.1059	12.0526	-0.50	-0.06
3	15	30	1.20	26.1836	26.2436	26.2050	-0.23	-0.08
3	18	3	0.50	0.0036	0.0041	0.0035	-15.64	1.12
3	18	3	0.90	0.4196	0.4288	0.4209	-2.19	-0.31
3	18	3	1.20	1.1510	1.1626	1.1562	-1.01	-0.46
3	18	10	0.50	0.0047	0.0054	0.0046	-15.99	1.28
3	18	10	0.90	1.7838	1.8208	1.7869	-2.07	-0.17
3	18	10	1.20	6.5804	6.6387	6.6021	-0.89	-0.33
3	50	25	0.50	0.0000	0.0000	0.0000	0.00	0.00
3	50	25	0.90	2.0619	2.1123	2.0604	-2.44	0.07
3	50	25	1.20	20.8700	20.9933	20.9074	-0.59	-0.18

Table 4.4: Expected Queue Length, Part 3

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	3	1	0.50	0.0544	0.0551	0.0535	-1.42	1.55
4	3	1	0.90	0.2013	0.2017	0.1996	-0.18	0.86
4	3	1	1.20	0.3159	0.3155	0.3139	0.12	0.60
4	3	10	0.50	0.1582	0.1625	0.1571	-2.74	0.68
4	3	10	0.90	2.6948	2.7114	2.6942	-0.62	0.02
4	3	10	0.99	4.1728	4.1896	4.1762	-0.40	-0.08
4	3	10	1.20	7.0486	7.0642	7.0599	-0.22	-0.16
4	3	25	0.50	0.1582	0.1626	0.1572	-2.74	0.68
4	3	25	0.90	4.3189	4.3391	4.3150	-0.47	0.09
4	3	25	0.99	10.8558	10.8760	10.8585	-0.19	-0.02
4	3	25	1.20	21.6514	21.6671	21.6641	-0.07	-0.06
4	3	50	0.50	0.1582	0.1626	0.1572	-2.74	0.68
4	3	50	0.90	4.6289	4.6496	4.6241	-0.45	0.10
4	3	50	0.99	20.7597	20.7660	20.7453	-0.03	0.07
4	3	50	1.20	46.6410	46.6567	46.6538	-0.03	-0.03
4	3	100	0.50	0.1582	0.1626	0.1572	-2.74	0.68
4	3	100	0.90	4.6384	4.6592	4.6336	-0.45	0.10
4	3	100	0.99	35.9004	35.8090	35.7839	0.25	0.32
4	3	100	1.20	96.6410	96.6567	96.6538	-0.02	-0.01
4	5	5	0.50	0.0869	0.0907	0.0862	-4.31	0.82
4	5	5	0.90	1.2660	1.2828	1.2706	-1.33	-0.36
4	5	5	1.20	2.7239	2.7450	2.7391	-0.77	-0.56
4	5	15	0.50	0.0906	0.0944	0.0897	-4.28	0.91
4	5	15	0.90	3.2856	3.3093	3.2844	-0.72	0.04
4	5	15	0.99	6.2569	6.2819	6.2628	-0.40	-0.10
4	5	15	1.20	11.6835	11.7079	11.7020	-0.21	-0.16
4	5	50	0.50	0.0906	0.0944	0.0897	-4.28	0.91
4	5	50	0.90	4.3419	4.3690	4.3384	-0.62	0.08
4	5	50	0.99	20.5118	20.5445	20.5185	-0.16	-0.03
4	5	50	1.20	46.5606	46.5852	46.5798	-0.05	-0.04

Table 4.5: Expected Queue Length, Part 4

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	10	5	0.50	0.0255	0.0276	0.0251	-8.53	1.26
4	10	5	0.90	1.0294	1.0509	1.0364	-2.09	-0.68
4	10	5	1.00	1.5621	1.5870	1.5750	-1.59	-0.82
4	10	5	1.20	2.5802	2.6090	2.6022	-1.12	-0.85
4	15	5	0.50	0.0083	0.0093	0.0081	-12.82	1.69
4	15	5	0.90	0.8816	0.9035	0.8886	-2.48	-0.79
4	15	5	1.20	2.4949	2.5250	2.5190	-1.21	-0.97
4	15	30	0.50	0.0087	0.0099	0.0086	-12.81	1.83
4	15	30	0.90	3.3590	3.4071	3.3557	-1.43	0.10
4	15	30	0.99	12.0185	12.0752	12.0310	-0.47	-0.10
4	15	30	1.20	26.3228	26.3706	26.3602	-0.18	-0.14
4	18	3	0.50	0.0035	0.0041	0.0035	-15.01	1.70
4	18	3	0.90	0.4171	0.4269	0.4194	-2.36	-0.57
4	18	3	1.20	1.1515	1.1636	1.1609	-1.05	-0.81
4	18	10	0.50	0.0046	0.0053	0.0045	-15.32	2.19
4	18	10	0.90	1.7598	1.7973	1.7652	-2.13	-0.31
4	18	10	1.20	6.6408	6.6891	6.6793	-0.73	-0.58
4	50	25	0.50	N/A	0.0000	0.0000	N/A	N/A
4	50	25	0.90	N/A	2.0477	1.9829	N/A	N/A
4	50	25	1.20	N/A	21.0302	21.0256	N/A	N/A

Table 4.6: Expected Queue Length, Part 5

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	3	1	0.50	0.0537	0.0541	0.0526	-0.76	2.03
5	3	1	0.90	0.2000	0.1997	0.1977	0.14	1.14
5	3	1	1.20	0.3146	0.3136	0.3121	0.32	0.79
5	3	10	0.50	0.1529	0.1563	0.1516	-2.23	0.88
5	3	10	0.90	2.6619	2.6755	2.6611	-0.51	0.03
5	3	10	0.99	4.1735	4.1882	4.1779	-0.35	-0.11
5	3	10	1.20	7.1059	7.1219	7.1207	-0.22	-0.21
5	3	25	0.50	0.1530	0.1564	0.1516	-2.24	0.88
5	3	25	0.90	4.1869	4.2024	4.1819	-0.37	0.12
5	3	25	0.99	10.8323	10.8496	10.8359	-0.16	-0.03
5	3	25	1.20	21.7567	21.7730	21.7732	-0.08	-0.08
5	3	50	0.50	0.1530	0.1564	0.1516	-2.24	0.88
5	3	50	0.90	4.4502	4.4658	4.4442	-0.35	0.14
5	3	50	0.99	20.6393	20.6380	20.6214	0.01	0.09
5	3	50	1.20	46.7492	46.7655	46.7658	-0.03	-0.04
5	3	100	0.50	0.1530	0.1564	0.1516	-2.24	0.88
5	3	100	0.90	4.4570	4.4725	4.4508	-0.35	0.14
5	3	100	0.99	35.4102	35.3010	35.2803	0.31	0.37
5	3	100	1.20	96.7492	96.7655	96.7657	-0.02	-0.02

Table 4.7. Expected Queue Length, Part 6

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	5	5	0.50	0.0847	0.0879	0.0838	-3.79	1.06
5	5	5	0.90	1.2594	1.2765	1.2655	-1.35	-0.48
5	5	5	1.20	2.7386	2.7625	2.7586	-0.87	-0.73
5	5	15	0.50	0.0879	0.0912	0.0869	-3.72	1.19
5	5	15	0.90	3.2240	3.2455	3.2224	-0.67	0.05
5	5	15	0.99	6.2557	6.2802	6.2636	-0.39	-0.12
5	5	15	1.20	11.7643	11.7911	11.7884	-0.23	-0.21
5	5	50	0.50	0.0879	0.0912	0.0869	-3.72	1.19
5	5	50	0.90	4.1766	4.2004	4.1719	-0.57	0.11
5	5	50	0.99	20.3972	20.4291	20.4059	-0.16	-0.04
5	5	50	1.20	46.6627	46.6898	46.6876	-0.06	-0.05
5	10	5	0.50	0.0250	0.0268	0.0245	-7.41	1.68
5	10	5	0.90	1.0238	1.0463	1.0332	-2.20	-0.92
5	10	5	1.00	1.5612	1.5888	1.5785	-1.77	-1.11
5	10	5	1.20	2.5898	2.6237	2.6192	-1.31	-1.14
5	15	5	0.50	0.0081	0.0091	0.0080	-11.17	2.21
5	15	5	0.90	0.8767	0.8999	0.8861	-2.64	-1.08
5	15	5	1.20	2.5018	2.5382	2.5340	-1.45	-1.29
5	15	30	0.50	0.0086	0.0095	0.0084	-11.06	2.56
5	15	30	0.90	3.2523	3.2972	3.2489	-1.38	0.10
5	15	30	0.99	11.9980	12.0541	12.0145	-0.47	-0.14
5	15	30	1.20	26.4057	26.4601	26.4545	-0.21	-0.19

Table 4.8: Expected Queue Length, Part 7

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
8	3	10	0.50	0.1450	0.1470	0.1432	-1.37	1.21
8	3	10	0.90	2.6073	2.6183	2.6062	-0.42	0.04
8	3	10	1.20	7.1970	7.2170	7.2180	-0.28	-0.29
8	5	10	0.50	0.0839	0.0858	0.0825	-2.31	1.66
8	5	10	0.90	2.4077	2.4243	2.4093	-0.69	-0.07
8	5	10	1.20	7.0944	7.1266	7.1266	-0.45	-0.45
10	3	2	0.50	0.0943	0.0957	0.0937	-1.49	0.60
10	3	2	0.90	0.4667	0.4747	0.4720	-1.72	-1.14
10	3	2	1.20	0.7924	0.8052	0.8040	-1.61	-1.46
10	3	10	0.50	0.1423	0.1437	0.1404	-0.95	1.33
10	3	10	0.90	2.5875	2.5969	2.5864	-0.36	0.04
10	3	10	1.20	7.2289	7.2508	7.2521	-0.30	-0.32
10	5	10	0.50	0.0825	0.0838	0.0810	-1.59	1.82
10	5	10	0.90	2.3906	2.4051	2.3925	-0.61	-0.08
10	5	10	1.20	7.1240	7.1590	7.1598	-0.49	-0.50
10	10	5	0.50	0.0240	0.0248	0.0233	-3.21	2.71
10	10	5	0.90	1.0101	1.0343	1.0259	-2.39	-1.56
10	10	5	1.20	2.6097	2.6583	2.6576	-1.86	-1.84
15	3	10	0.50	0.1387	0.1390	0.1367	-0.22	1.46
15	3	10	0.90	2.5597	2.5661	2.5586	-0.25	0.04
15	3	10	1.20	7.2726	7.2974	7.2990	-0.34	-0.36
15	5	10	0.50	0.0807	0.0810	0.0791	-0.45	2.01
15	5	10	0.90	2.3662	2.3779	2.3688	-0.49	-0.11
15	5	10	1.20	7.1646	7.2041	7.2053	-0.55	-0.57
20	3	10	0.50	0.1368	0.1366	0.1348	0.18	1.51
20	3	10	0.90	2.5451	2.5502	2.5441	-0.20	0.04
20	3	10	1.20	7.2951	7.3216	7.3231	-0.36	-0.38

Table 4.9: Expected Virtual Delay, Part 1

k	n	q	ρ	Delay Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
3	3	1	0.50	0.7258	0.7421	0.7288	-2.24	-0.41
3	3	1	0.90	1.9138	1.9325	1.9223	-0.98	-0.45
3	3	1	1.20	2.6236	2.6413	2.6350	-0.68	-0.44
3	3	10	0.50	1.3299	1.3589	1.3301	-2.18	-0.01
3	3	10	0.90	11.7035	11.7541	11.7044	-0.43	-0.01
3	3	10	0.99	16.7486	16.7968	16.7564	-0.29	-0.05
3	3	10	1.20	26.1824	26.2257	26.2041	-0.17	-0.08
3	3	25	0.50	1.3303	1.3593	1.3305	-2.18	-0.01
3	3	25	0.90	17.7931	17.8504	17.7869	-0.32	0.03
3	3	25	0.99	39.4030	39.4575	39.4088	-0.14	-0.01
3	3	25	1.20	74.6444	74.6887	74.6691	-0.06	-0.03
3	3	50	0.50	1.3303	1.3593	1.3305	-2.18	-0.01
3	3	50	0.90	19.1258	19.1840	19.1173	-0.30	0.04
3	3	50	0.99	72.9962	73.0167	72.9624	-0.03	0.05
3	3	50	1.20	157.9232	157.9674	157.9479	-0.03	-0.02
3	3	100	0.50	1.3303	1.3593	1.3305	-2.18	-0.01
3	3	100	0.90	19.1783	19.2368	19.1700	-0.30	0.04
3	3	100	0.99	125.3812	125.1208	125.0588	0.21	0.26
3	3	100	1.20	324.5898	324.6338	324.6143	-0.01	-0.01
3	5	5	0.50	0.4319	0.4479	0.4320	-3.71	-0.03
3	5	5	0.90	3.7633	3.8010	3.7714	-1.00	-0.22
3	5	5	1.20	7.0042	7.0429	7.0231	-0.55	-0.27
3	5	15	0.50	0.4429	0.4593	0.4429	-3.69	0.02
3	5	15	0.90	8.2273	8.2758	8.2272	-0.59	0.00
3	5	15	0.99	14.2500	14.2990	14.2576	-0.34	-0.05
3	5	15	1.20	24.9277	24.9730	24.9489	-0.18	-0.09
3	5	50	0.50	0.4429	0.4593	0.4429	-3.69	0.02
3	5	50	0.90	10.7428	10.7976	10.7410	-0.51	0.02
3	5	50	0.99	43.2591	43.3182	43.2675	-0.14	-0.02
3	5	50	1.20	94.6130	94.6590	94.6352	-0.05	-0.02

Table 4.10: Expected Virtual Delay, Part 2

k	n	q	ρ	Delay Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
3	10	5	0.50	0.0606	0.0654	0.0605	-8.06	0.08
3	10	5	0.90	1.5251	1.5519	1.5311	-1.76	-0.39
3	10	5	1.00	2.1786	2.2076	2.1881	-1.33	-0.44
3	10	5	1.20	3.3294	3.3602	3.3436	-0.92	-0.43
3	15	5	0.50	0.0128	0.0144	0.0127	-12.75	0.23
3	15	5	0.90	0.8680	0.8874	0.8722	-2.24	-0.49
3	15	5	1.20	2.1491	2.1734	2.1599	-1.13	-0.51
3	15	30	0.50	0.0132	0.0149	0.0132	-12.77	0.30
3	15	30	0.90	2.7452	2.7782	2.7443	-1.21	0.03
3	15	30	0.99	8.6120	8.6538	8.6170	-0.49	-0.06
3	15	30	1.20	18.0667	18.1067	18.0810	-0.22	-0.08
3	18	3	0.50	0.0049	0.0056	0.0048	-15.46	0.21
3	18	3	0.90	0.3952	0.4046	0.3974	-2.37	-0.55
3	18	3	1.20	0.9622	0.9735	0.9677	-1.17	-0.57
3	18	10	0.50	0.0057	0.0066	0.0057	-15.82	0.35
3	18	10	0.90	1.2624	1.2875	1.2648	-1.99	-0.19
3	18	10	1.20	4.1407	4.1741	4.1530	-0.81	-0.30
3	50	25	0.50	0.0000	0.0000	0.0000	0.00	0.00
3	50	25	0.90	0.4814	0.4934	0.4813	-2.48	0.03
3	50	25	1.20	4.3571	4.3818	4.3646	-0.57	-0.17

Table 4.11: Expected Virtual Delay, Part 3

k	n	q	ρ	Delay Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	3	1	0.50	0.7228	0.7425	0.7283	-2.72	-0.77
4	3	1	0.90	1.9131	1.9394	1.9287	-1.37	-0.82
4	3	1	1.20	2.6270	2.6530	2.6477	-0.99	-0.79
4	3	10	0.50	1.2973	1.3308	1.2980	-2.58	-0.05
4	3	10	0.90	11.5507	11.6168	11.5522	-0.57	-0.01
4	3	10	0.99	16.7757	16.8373	16.7893	-0.37	-0.08
4	3	10	1.20	26.4936	26.5464	26.5317	-0.20	-0.14
4	3	25	0.50	1.2975	1.3310	1.2982	-2.58	-0.05
4	3	25	0.90	17.0937	17.1704	17.0832	-0.45	0.06
4	3	25	0.99	39.3009	39.3711	39.3107	-0.18	-0.02
4	3	25	1.20	75.2261	75.2785	75.2686	-0.07	-0.06
4	3	50	0.50	1.2975	1.3310	1.2982	-2.58	-0.05
4	3	50	0.90	18.1367	18.2150	18.1233	-0.43	0.07
4	3	50	0.99	72.4043	72.4267	72.3565	-0.03	0.07
4	3	50	1.20	158.5256	158.5780	158.5682	-0.03	-0.03
4	3	100	0.50	1.2975	1.3310	1.2982	-2.58	-0.05
4	3	100	0.90	18.1686	18.2471	18.1551	-0.43	0.07
4	3	100	0.99	122.9165	122.6126	122.5280	0.25	0.32
4	3	100	1.20	325.1922	325.2445	325.2348	-0.02	-0.01
4	5	5	0.50	0.4233	0.4413	0.4236	-4.25	-0.08
4	5	5	0.90	3.7563	3.8005	3.7705	-1.18	-0.38
4	5	5	1.20	7.0675	7.1131	7.1008	-0.65	-0.47
4	5	15	0.50	0.4324	0.4507	0.4324	-4.23	-0.01
4	5	15	0.90	8.0419	8.0972	8.0418	-0.69	0.00
4	5	15	0.99	14.2607	14.3145	14.2740	-0.38	-0.09
4	5	15	1.20	25.1909	25.2397	25.2280	-0.19	-0.15
4	5	50	0.50	0.4324	0.4507	0.4324	-4.23	-0.01
4	5	50	0.90	10.1933	10.2551	10.1883	-0.61	0.05
4	5	50	0.99	42.9238	42.9906	42.9375	-0.16	-0.03
4	5	50	1.20	94.9546	95.0037	94.9929	-0.05	-0.04

Table 4.12: Expected Virtual Delay, Part 4

k	n	q	ρ	Delay Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	10	5	0.50	0.0595	0.0645	0.0594	-8.42	0.13
4	10	5	0.90	1.5214	1.5507	1.5322	-1.92	-0.71
4	10	5	1.00	2.1853	2.2168	2.2025	-1.44	-0.79
4	10	5	1.20	3.3534	3.3862	3.3789	-0.98	-0.76
4	15	5	0.50	0.0126	0.0142	0.0125	-12.62	0.48
4	15	5	0.90	0.8652	0.8857	0.8729	-2.37	-0.89
4	15	5	1.20	2.1617	2.1857	2.1813	-1.11	-0.91
4	15	30	0.50	0.0130	0.0146	0.0129	-12.62	0.62
4	15	30	0.90	2.6307	2.6673	2.6294	-1.39	0.05
4	15	30	0.99	8.5969	8.6362	8.6057	-0.46	-0.10
4	15	30	1.20	18.1596	18.1915	18.1845	-0.18	-0.14
4	18	3	0.50	0.0048	0.0055	0.0048	-14.97	0.42
4	18	3	0.90	0.3942	0.4041	0.3982	-2.51	-1.00
4	18	3	1.20	0.9643	0.9757	0.9740	-1.18	-1.01
4	18	10	0.50	0.0056	0.0065	0.0056	-15.18	0.71
4	18	10	0.90	1.2505	1.2757	1.2548	-2.01	-0.34
4	18	10	1.20	4.1772	4.2047	4.1991	-0.66	-0.52
4	50	25	0.50	N/A	0.0000	0.0000	N/A	N/A
4	50	25	0.90	N/A	0.4800	0.4656	N/A	N/A
4	50	25	1.20	N/A	4.3892	4.3883	N/A	N/A

Table 4.13: Expected Virtual Delay, Part 5

k	n	q	ρ	Delay Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	3	1	0.50	0.7205	0.7411	0.7280	-2.86	-1.05
5	3	1	0.90	1.9120	1.9419	1.9330	-1.56	-1.10
5	3	1	1.20	2.6286	2.6595	2.6562	-1.18	-1.05
5	3	10	0.50	1.2772	1.3080	1.2785	-2.41	-0.10
5	3	10	0.90	11.4482	11.5043	11.4501	-0.49	-0.02
5	3	10	0.99	16.7907	16.8453	16.8085	-0.33	-0.11
5	3	10	1.20	26.6911	26.7451	26.7411	-0.20	-0.19
5	3	25	0.50	1.2774	1.3081	1.2787	-2.41	-0.10
5	3	25	0.90	16.6531	16.7142	16.6399	-0.37	0.08
5	3	25	0.99	39.2286	39.2891	39.2419	-0.15	-0.03
5	3	25	1.20	75.5773	75.6319	75.6325	-0.07	-0.07
5	3	50	0.50	1.2774	1.3081	1.2787	-2.41	-0.10
5	3	50	0.90	17.5391	17.6006	17.5225	-0.35	0.09
5	3	50	0.99	72.0059	72.0031	71.9469	0.00	0.08
5	3	50	1.20	158.8861	158.9406	158.9414	-0.03	-0.03
5	3	100	0.50	1.2774	1.3081	1.2787	-2.41	-0.10
5	3	100	0.90	17.5618	17.6229	17.5448	-0.35	0.10
5	3	100	0.99	121.2834	120.9207	120.8507	0.30	0.36
5	3	100	1.20	325.5528	325.6072	325.6080	-0.02	-0.02

Table 4.14: Expected Virtual Delay, Part 6

k	n	q	ρ	Delay Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	5	5	0.50	0.4179	0.4347	0.4184	-4.03	-0.13
5	5	5	0.90	3.7508	3.7961	3.7693	-1.21	-0.49
5	5	5	1.20	7.1077	7.1593	7.1515	-0.73	-0.62
5	5	15	0.50	0.4259	0.4429	0.4261	-3.99	-0.04
5	5	15	0.90	7.9202	7.9716	7.9202	-0.65	0.00
5	5	15	0.99	14.2651	14.3176	14.2825	-0.37	-0.12
5	5	15	1.20	25.3539	25.4078	25.4024	-0.21	-0.19
5	5	50	0.50	0.4259	0.4429	0.4261	-3.99	-0.04
5	5	50	0.90	9.8607	9.9163	9.8541	-0.56	0.07
5	5	50	0.99	42.6968	42.7622	42.7147	-0.15	-0.04
5	5	50	1.20	95.1587	95.2129	95.2086	-0.06	-0.05
5	10	5	0.50	0.0589	0.0634	0.0588	-7.75	0.17
5	10	5	0.90	1.5183	1.5493	1.5328	-2.04	-0.96
5	10	5	1.00	2.1889	2.2239	2.2119	-1.59	-1.05
5	10	5	1.20	3.3682	3.4067	3.4022	-1.14	-1.01
5	15	5	0.50	0.0125	0.0139	0.0124	-11.62	0.48
5	15	5	0.90	0.8630	0.8850	0.8734	-2.55	-1.20
5	15	5	1.20	2.1692	2.1982	2.1953	-1.33	-1.20
5	15	30	0.50	0.0129	0.0143	0.0128	-11.43	0.78
5	15	30	0.90	2.5586	2.5932	2.5576	-1.35	0.04
5	15	30	0.99	8.5850	8.6239	8.5966	-0.45	-0.13
5	15	30	1.20	18.2148	18.2511	18.2474	-0.20	-0.18

Table 4.15: Expected Virtual Delay, Part 7

k	n	q	ρ	Delay Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
8	3	10	0.50	1.2462	1.2720	1.2491	-2.07	-0.23
8	3	10	0.90	11.2766	11.3246	11.2797	-0.43	-0.03
8	3	10	1.20	27.0040	27.0714	27.0747	-0.25	-0.26
8	5	10	0.50	0.4157	0.4295	0.4164	-3.30	-0.17
8	5	10	0.90	6.2374	6.2800	6.2462	-0.68	-0.14
8	5	10	1.20	15.9904	16.0555	16.0556	-0.41	-0.41
10	3	2	0.50	0.9951	1.0225	1.0087	-2.76	-1.37
10	3	2	0.90	3.2941	3.3566	3.3466	-1.90	-1.59
10	3	2	1.20	4.8224	4.8975	4.8965	-1.56	-1.54
10	3	10	0.50	1.2355	1.2591	1.2392	-1.91	-0.30
10	3	10	0.90	11.2139	11.2569	11.2177	-0.38	-0.03
10	3	10	1.20	27.1131	27.1869	27.1915	-0.27	-0.29
10	5	10	0.50	0.4122	0.4242	0.4132	-2.91	-0.24
10	5	10	0.90	6.2049	6.2436	6.2150	-0.62	-0.16
10	5	10	1.20	16.0517	16.1225	16.1240	-0.44	-0.45
10	10	5	0.50	0.0575	0.0603	0.0574	-4.94	0.17
10	10	5	0.90	1.5096	1.5439	1.5337	-2.27	-1.60
10	10	5	1.20	3.3994	3.4541	3.4539	-1.61	-1.61
15	3	10	0.50	1.2208	1.2404	1.2260	-1.61	-0.43
15	3	10	0.90	11.1253	11.1586	11.1305	-0.30	-0.05
15	3	10	1.20	27.2626	27.3462	27.3516	-0.31	-0.33
15	5	10	0.50	0.4074	0.4167	0.4089	-2.28	-0.37
15	5	10	0.90	6.1585	6.1914	6.1711	-0.53	-0.20
15	5	10	1.20	16.1357	16.2155	16.2179	-0.49	-0.51
20	3	10	0.50	1.2131	1.2307	1.2194	-1.45	-0.52
20	3	10	0.90	11.0785	11.1074	11.0849	-0.26	-0.06
20	3	10	1.20	27.3393	27.4286	27.4337	-0.33	-0.35

The three remaining aggregate performance measures: expected waiting time in the system, expected number of customers in the system, and expected number of busy servers, are approximated very well using either heuristic. As seen in Tables 4.16 through 4.36, the heuristic solutions are always within 1% of the exact solution. The expected number of busy servers is approximated particularly well, with the results for both heuristics being, in most cases, identical to 3 decimal places with the exact values.

Notice that for both L_q and $E[Delay]$, the performance of the heuristics is worst for low utilization ratios. We suspect that for low utilization ratios, the probabilities of having more customers than the number of servers ($m > n$) are small compared with the probabilities of having few customers in the system, therefore, the percentage differences are larger but the absolute differences are small. For example, in Table 4.2, the $M/E_3/10/10+5$ system has L_q = 0.0263, 0.0284 and 0.0261 for the exact, ELP and ELC, respectively, giving a percentage difference of -8.3% and 0.72% for ELP and ELC, respectively, with respect to the exact. The absolute differences are 0.0021 and 0.0002 for ELP and ELC, respectively. In contrast to L_q and $E[Delay]$, the aggregate measures W, L and $E[Busy]$ do not worsen for any particular changes in the system parameters or utilization ratios. This may be because we use all occupancy probabilities in evaluating W, L and $E[Busy]$, instead of a subset of the occupancy probabilities required to compute L_q and $E[Delay]$.

For all aggregate measures, the results obtained using the heuristic solution techniques are sometimes larger, sometimes smaller than the exact solutions. Therefore, neither ELP nor ELC provides an upper or a lower bound for the exact results.

The examples and results presented so far in this section show clearly that ELC dominates ELP. The heuristic technique ELC provides a superb approximation of the exact results. In ELC, only 5% of the results differed more than 1% from the exact results; no values differ more than 3%. Using ELP, 18% of the results differ more than 1% and 6% of them more than 3% from the exact values. If the computing times for the heuristic solution techniques are similar, then the heuristic of choice would be ELC. In the paragraphs below, we provide an analysis of the number of Chapman-

Table 4.16: Expected Waiting Time in System, Part 1

k	n	q	ρ	Waiting Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
3	3	1	0.50	10.3909	10.3971	10.3873	-0.06	0.03
3	3	1	0.90	10.9459	10.9483	10.9401	-0.02	0.05
3	3	1	1.20	11.2942	11.2931	11.2877	0.01	0.06
3	3	10	0.50	11.1125	11.1380	11.1082	-0.23	0.04
3	3	10	0.90	20.3850	20.4334	20.3834	-0.24	0.01
3	3	10	0.99	24.8242	24.8710	24.8302	-0.19	-0.02
3	3	10	1.20	33.3901	33.4329	33.4109	-0.13	-0.06
3	3	25	0.50	11.1131	11.1387	11.1088	-0.23	0.04
3	3	25	0.90	26.8017	26.8575	26.7932	-0.21	0.03
3	3	25	0.99	47.4306	47.4845	47.4356	-0.11	-0.01
3	3	25	1.20	81.5987	81.6430	81.6234	-0.05	-0.03
3	3	50	0.50	11.1131	11.1387	11.1088	-0.23	0.04
3	3	50	0.90	28.2403	28.2971	28.2294	-0.20	0.04
3	3	50	0.99	81.1113	81.1299	81.0754	-0.02	0.04
3	3	50	1.20	164.8676	164.9119	164.8924	-0.03	-0.02
3	3	100	0.50	11.1131	11.1387	11.1088	-0.23	0.04
3	3	100	0.90	28.2978	28.3548	28.2870	-0.20	0.04
3	3	100	0.99	133.7097	133.4420	133.3799	0.20	0.25
3	3	100	1.20	331.5342	331.5780	331.5585	-0.01	-0.01
3	5	5	0.50	10.3623	10.3764	10.3607	-0.14	0.02
3	5	5	0.90	13.0018	13.0357	13.0073	-0.26	-0.04
3	5	5	1.20	15.5983	15.6342	15.6150	-0.23	-0.11
3	5	15	0.50	10.3799	10.3945	10.3779	-0.14	0.02
3	5	15	0.90	17.5741	17.6210	17.5724	-0.27	0.01
3	5	15	0.99	23.0950	23.1429	23.1016	-0.21	-0.03
3	5	15	1.20	33.1500	33.1952	33.1711	-0.14	-0.06
3	5	50	0.50	10.3799	10.3945	10.3779	-0.14	0.02
3	5	50	0.90	20.2565	20.3104	20.2533	-0.27	0.02
3	5	50	0.99	52.1390	52.1980	52.1472	-0.11	-0.02
3	5	50	1.20	102.7796	102.8257	102.8018	-0.04	-0.02

Table 4.17: Expected Waiting Time in System, Part 2

k	n	q	ρ	Waiting Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
3	10	5	0.50	10.0525	10.0569	10.0521	-0.04	0.00
3	10	5	0.90	11.2109	11.2337	11.2150	-0.20	-0.04
3	10	5	1.00	11.7159	11.7405	11.7231	-0.21	-0.06
3	10	5	1.20	12.6447	12.6711	12.6565	-0.21	-0.09
3	15	5	0.50	10.0113	10.0128	10.0112	-0.01	0.00
3	15	5	0.90	10.6875	10.7033	10.6902	-0.15	-0.03
3	15	5	1.20	11.7018	11.7214	11.7104	-0.17	-0.07
3	15	30	0.50	10.0120	10.0136	10.0119	-0.02	0.00
3	15	30	0.90	12.6157	12.6476	12.6142	-0.25	0.01
3	15	30	0.99	18.2410	18.2823	18.2458	-0.23	-0.03
3	15	30	1.20	27.4561	27.4961	27.4704	-0.15	-0.05
3	18	3	0.50	10.0040	10.0046	10.0039	-0.01	0.00
3	18	3	0.90	10.2757	10.2819	10.2765	-0.06	-0.01
3	18	3	1.20	10.6710	10.6775	10.6737	-0.06	-0.03
3	18	10	0.50	10.0052	10.0061	10.0052	-0.01	0.00
3	18	10	0.90	11.1186	11.1418	11.1204	-0.21	-0.02
3	18	10	1.20	13.6777	13.7098	13.6896	-0.23	-0.09
3	50	25	0.50	10.0000	10.0000	10.0000	0.00	-0.00
3	50	25	0.90	10.4586	10.4698	10.4583	-0.11	0.00
3	50	25	1.20	14.1743	14.1989	14.1818	-0.17	-0.05

Table 4.18: Expected Waiting Time in System, Part 3

k	n	q	ρ	Waiting Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	3	1	0.50	10.3831	10.3889	10.3769	-0.06	0.06
4	3	1	0.90	10.9337	10.9357	10.9236	-0.02	0.09
4	3	1	1.20	11.2824	11.2801	11.2711	0.02	0.10
4	3	10	0.50	11.0545	11.0833	11.0473	-0.26	0.06
4	3	10	0.90	20.1745	20.2380	20.1716	-0.31	0.01
4	3	10	0.99	24.7866	24.8467	24.7971	-0.24	-0.04
4	3	10	1.20	33.6587	33.7114	33.6955	-0.16	-0.11
4	3	25	0.50	11.0548	11.0837	11.0477	-0.26	0.06
4	3	25	0.90	26.0174	26.0924	26.0028	-0.29	0.06
4	3	25	0.99	47.2609	47.3307	47.2695	-0.15	-0.02
4	3	25	1.20	82.1767	82.2292	82.2192	-0.06	-0.05
4	3	50	0.50	11.0549	11.0837	11.0477	-0.26	0.06
4	3	50	0.90	27.1443	27.2210	27.1266	-0.28	0.07
4	3	50	0.99	80.4508	80.4707	80.4000	-0.02	0.06
4	3	50	1.20	165.4700	165.5224	165.5127	-0.03	-0.03
4	3	100	0.50	11.0548	11.0837	11.0477	-0.26	0.06
4	3	100	0.90	27.1792	27.2561	27.1615	-0.28	0.07
4	3	100	0.99	131.1705	130.8585	130.7736	0.24	0.30
4	3	100	1.20	332.1367	332.1888	332.1790	-0.02	-0.01
4	5	5	0.50	10.3479	10.3629	10.3450	-0.15	0.03
4	5	5	0.90	12.9696	13.0094	12.9793	-0.31	-0.07
4	5	5	1.20	15.6284	15.6717	15.6581	-0.28	-0.19
4	5	15	0.50	10.3622	10.3778	10.3590	-0.15	0.03
4	5	15	0.90	17.3542	17.4072	17.3512	-0.31	0.02
4	5	15	0.99	23.0675	23.1201	23.0792	-0.23	-0.05
4	5	15	1.20	33.4004	33.4491	33.4372	-0.15	-0.11
4	5	50	0.50	10.3623	10.3778	10.3590	-0.15	0.03
4	5	50	0.90	19.6488	19.7091	19.6411	-0.31	0.04
4	5	50	0.99	51.7632	51.8299	51.7767	-0.13	-0.03
4	5	50	1.20	103.1212	103.1703	103.1596	-0.05	-0.04

Table 4.19: Expected Waiting Time in System, Part 4

k	n	q	ρ	Waiting Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	10	5	0.50	10.0509	10.0552	10.0503	-0.04	0.01
4	10	5	0.90	11.1990	11.2242	11.2065	-0.23	-0.07
4	10	5	1.00	11.7109	11.7382	11.7240	-0.23	-0.11
4	10	5	1.20	12.6551	12.6843	12.6766	-0.23	-0.17
4	15	5	0.50	10.0110	10.0124	10.0108	-0.01	0.00
4	15	5	0.90	10.6811	10.6981	10.6861	-0.16	-0.05
4	15	5	1.20	11.7071	11.7272	11.7227	-0.17	-0.13
4	15	30	0.50	10.0117	10.0131	10.0114	-0.01	0.00
4	15	30	0.90	12.4893	12.5250	12.4869	-0.29	0.02
4	15	30	0.99	18.2138	18.2526	18.2221	-0.21	-0.05
4	15	30	1.20	27.5488	27.5807	27.5737	-0.12	-0.09
4	18	3	0.50	10.0039	10.0045	10.0039	-0.01	0.00
4	18	3	0.90	10.2738	10.2805	10.2752	-0.06	-0.01
4	18	3	1.20	10.6707	10.6775	10.6755	-0.06	-0.04
4	18	10	0.50	10.0051	10.0059	10.0050	-0.01	0.00
4	18	10	0.90	11.1021	11.1258	11.1053	-0.21	-0.03
4	18	10	1.20	13.7084	13.7353	13.7296	-0.20	-0.15
4	50	25	0.50	N/A	10.0000	10.0000	N/A	N/A
4	50	25	0.90	N/A	10.4554	10.4410	N/A	N/A
4	50	25	1.20	N/A	14.2062	14.2053	N/A	N/A

Table 4.20: Expected Waiting Time in System, Part 5

k	n	q	ρ	Waiting Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	3	1	0.50	10.3784	10.3814	10.3703	-0.03	0.08
5	3	1	0.90	10.9259	10.9243	10.9128	0.01	0.12
5	3	1	1.20	11.2750	11.2690	11.2602	0.05	0.13
5	3	10	0.50	11.0195	11.0422	11.0105	-0.21	0.08
5	3	10	0.90	20.0368	20.0883	20.0329	-0.26	0.02
5	3	10	0.99	24.7619	24.8139	24.7757	-0.21	-0.06
5	3	10	1.20	33.8318	33.8852	33.8802	-0.16	-0.14
5	3	25	0.50	11.0198	11.0425	11.0108	-0.21	0.08
5	3	25	0.90	25.5247	25.5822	25.5061	-0.23	0.07
5	3	25	0.99	47.1478	47.2074	47.1596	-0.13	-0.03
5	3	25	1.20	82.5263	82.5809	82.5815	-0.07	-0.07
5	3	50	0.50	11.0198	11.0425	11.0108	-0.21	0.08
5	3	50	0.90	26.4825	26.5404	26.4602	-0.22	0.08
5	3	50	0.99	80.0109	80.0049	79.9483	0.01	0.08
5	3	50	1.20	165.8306	165.8851	165.8859	-0.03	-0.03
5	3	100	0.50	11.0198	11.0425	11.0108	-0.21	0.08
5	3	100	0.90	26.5074	26.5648	26.4846	-0.22	0.09
5	3	100	0.99	129.4921	129.1204	129.0503	0.29	0.34
5	3	100	1.20	332.4972	332.5515	332.5523	-0.02	-0.02

Table 4.21: Expected Waiting Time in System, Part 6

k	n	q	ρ	Waiting Time (seconds) Exact	ELP	ELC	% Difference ELP	ELC
5	5	5	0.50	10.3390	10.3518	10.3353	-0.12	0.04
5	5	5	0.90	12.9484	12.9883	12.9611	-0.31	-0.10
5	5	5	1.20	15.6479	15.6968	15.6873	-0.31	-0.25
5	5	15	0.50	10.3516	10.3647	10.3475	-0.13	0.04
5	5	15	0.90	17.2113	17.2596	17.2075	-0.28	0.02
5	5	15	0.99	23.0485	23.0996	23.0639	-0.22	-0.07
5	5	15	1.20	33.5566	33.6102	33.6047	-0.16	-0.14
5	5	50	0.50	10.3516	10.3647	10.3475	-0.13	0.04
5	5	50	0.90	19.2814	19.3344	19.2710	-0.27	0.05
5	5	50	0.99	51.5118	51.5770	51.5293	-0.13	-0.03
5	5	50	1.20	103.3254	103.3796	103.3752	-0.05	-0.05
5	10	5	0.50	10.0500	10.0537	10.0491	-0.04	0.01
5	10	5	0.90	11.1908	11.2170	11.2009	-0.23	-0.09
5	10	5	1.00	11.7072	11.7371	11.7248	-0.26	-0.15
5	10	5	1.20	12.6614	12.6955	12.6902	-0.27	-0.23
5	15	5	0.50	10.0109	10.0121	10.0106	-0.01	0.00
5	15	5	0.90	10.6766	10.6945	10.6834	-0.17	-0.06
5	15	5	1.20	11.7102	11.7343	11.7309	-0.21	-0.18
5	15	30	0.50	10.0114	10.0127	10.0112	-0.01	0.00
5	15	30	0.90	12.4099	12.4434	12.4075	-0.27	0.02
5	15	30	0.99	18.1944	18.2328	18.2054	-0.21	-0.06
5	15	30	1.20	27.6039	27.6402	27.6365	-0.13	-0.12

Table 4.22: Expected Waiting Time in System, Part 7

k	n	q	ρ	Waiting Time (seconds)			% Difference	
				Exact	ELP	ELC	ELP	ELC
8	3	10	0.50	10.9666	10.9798	10.9548	-0.12	0.11
8	3	10	0.90	19.8114	19.8525	19.8061	-0.21	0.03
8	3	10	1.20	34.1096	34.1760	34.1784	-0.19	-0.20
8	5	10	0.50	10.3355	10.3433	10.3299	-0.08	0.05
8	5	10	0.90	15.4323	15.4696	15.4353	-0.24	-0.02
8	5	10	1.20	24.2557	24.3200	24.3195	-0.27	-0.26
10	3	2	0.50	10.6403	10.6495	10.6358	-0.09	0.04
10	3	2	0.90	11.9864	12.0178	12.0039	-0.26	-0.15
10	3	2	1.20	12.9446	12.9870	12.9795	-0.33	-0.27
10	3	10	0.50	10.9487	10.9577	10.9361	-0.08	0.11
10	3	10	0.90	19.7305	19.7654	19.7250	-0.18	0.03
10	3	10	1.20	34.2075	34.2802	34.2839	-0.21	-0.22
10	5	10	0.50	10.3301	10.3353	10.3241	-0.05	0.06
10	5	10	0.90	15.3903	15.4230	15.3939	-0.21	-0.02
10	5	10	1.20	24.3105	24.3803	24.3813	-0.29	-0.29
10	10	5	0.50	10.0480	10.0495	10.0467	-0.02	0.01
10	10	5	0.90	11.1715	11.1989	11.1886	-0.25	-0.15
10	10	5	1.20	12.6743	12.7227	12.7213	-0.38	-0.37
15	3	10	0.50	10.9246	10.9266	10.9111	-0.02	0.12
15	3	10	0.90	19.6174	19.6410	19.6121	-0.12	0.03
15	3	10	1.20	34.3425	34.4247	34.4295	-0.24	-0.25
15	5	10	0.50	10.3227	10.3241	10.3162	-0.01	0.06
15	5	10	0.90	15.3311	15.3569	15.3362	-0.17	-0.03
15	5	10	1.20	24.3858	24.4645	24.4666	-0.32	-0.33
20	3	10	0.50	10.9123	10.9107	10.8986	0.01	0.13
20	3	10	0.90	19.5583	19.5766	19.5535	-0.09	0.02
20	3	10	1.20	34.4121	34.4999	34.5045	-0.26	-0.27

Table 4.23: Expected Number of Customers in System, Part 1

k	n	q	ρ	Number of Customers Exact	ELP	ELC	% Difference ELP	ELC
3	3	1	0.50	1.4723	1.4719	1.4726	0.03	-0.02
3	3	1	0.90	2.3542	2.3534	2.3559	0.03	-0.07
3	3	1	1.20	2.7736	2.7741	2.7765	-0.02	-0.10
3	3	10	0.50	1.6669	1.6707	1.6662	-0.23	0.04
3	3	10	0.90	5.3869	5.3996	5.3868	-0.23	0.00
3	3	10	0.99	6.9845	6.9975	6.9866	-0.19	-0.03
3	3	10	1.20	9.9337	9.9465	9.9402	-0.13	-0.07
3	3	25	0.50	1.6670	1.6708	1.6663	-0.23	0.04
3	3	25	0.90	7.2240	7.2390	7.2217	-0.21	0.03
3	3	25	0.99	13.7988	13.8144	13.8004	-0.11	-0.01
3	3	25	1.20	24.4766	24.4899	24.4840	-0.05	-0.03
3	3	50	0.50	1.6670	1.6708	1.6663	-0.23	0.04
3	3	50	0.90	7.6246	7.6399	7.6217	-0.20	0.04
3	3	50	0.99	23.8834	23.8891	23.8731	-0.02	0.04
3	3	50	1.20	49.4603	49.4736	49.4677	-0.03	-0.02
3	3	100	0.50	1.6670	1.6708	1.6663	-0.23	0.04
3	3	100	0.90	7.6404	7.6558	7.6375	-0.20	0.04
3	3	100	0.99	39.6021	39.5236	39.5052	0.20	0.24
3	3	100	1.20	99.4603	99.4735	99.4676	-0.01	-0.01
3	5	5	0.50	2.5886	2.5920	2.5882	-0.13	0.01
3	5	5	0.90	5.5249	5.5389	5.5285	-0.25	-0.07
3	5	5	1.20	7.5259	7.5438	7.5354	-0.24	-0.12
3	5	15	0.50	2.5950	2.5986	2.5945	-0.14	0.02
3	5	15	0.90	7.8424	7.8633	7.8418	-0.27	0.01
3	5	15	0.99	4.7781	4.7781	4.7783	0.00	-0.00
3	5	15	1.20	16.5440	16.5666	16.5546	-0.14	-0.06
3	5	50	0.50	2.5950	2.5986	2.5945	-0.14	0.02
3	5	50	0.90	9.1151	9.1394	9.1137	-0.27	0.02
3	5	50	0.99	25.5892	25.6181	25.5933	-0.11	-0.02
3	5	50	1.20	51.3898	51.4128	51.4009	-0.04	-0.02

Table 4.24: Expected Number of Customers in System, Part 2

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
3	10	5	0.50	5.0249	5.0270	5.0248	-0.04	0.00
3	10	5	0.90	9.6049	9.6238	9.6113	-0.20	-0.07
3	10	5	1.00	10.6687	10.6920	10.6789	-0.22	-0.10
3	10	5	1.20	12.2615	12.2908	12.2763	-0.24	-0.12
3	15	5	0.50	7.5077	7.5088	7.5077	-0.01	0.00
3	15	5	0.90	13.8120	13.8318	13.8201	-0.14	-0.06
3	15	5	1.20	17.0746	17.1114	17.0927	-0.22	-0.11
3	15	30	0.50	7.5090	7.5102	7.5089	-0.02	0.00
3	15	30	0.90	17.0205	17.0636	17.0185	-0.25	0.01
3	15	30	0.99	26.6620	26.7226	26.6694	-0.23	-0.03
3	15	30	1.20	41.1833	41.2432	41.2047	-0.15	-0.05
3	18	3	0.50	9.0010	9.0012	9.0010	-0.00	-0.00
3	18	3	0.90	15.6368	15.6394	15.6449	-0.02	-0.05
3	18	3	1.20	18.3028	18.3236	18.3180	-0.11	-0.08
3	18	10	0.50	9.0047	9.0054	9.0046	-0.01	0.00
3	18	10	0.90	17.7309	17.7682	17.7355	-0.21	-0.03
3	18	10	1.20	24.4731	24.5336	24.4956	-0.25	-0.09
3	50	25	0.50	24.9990	24.9996	25.0000	-0.00	-0.00
3	50	25	0.90	47.0191	47.0703	47.0182	-0.11	0.00
3	50	25	1.20	70.8665	70.9901	70.9039	-0.17	-0.05

Table 4.25: Expected Number of Customers in System, Part 3

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	3	1	0.50	1.4728	1.4724	1.4732	0.03	-0.03
4	3	1	0.90	2.3577	2.3571	2.3607	0.03	-0.13
4	3	1	1.20	2.7788	2.7798	2.7838	-0.04	-0.18
4	3	10	0.50	1.6582	1.6625	1.6571	-0.26	0.06
4	3	10	0.90	5.3433	5.3598	5.3429	-0.31	0.01
4	3	10	0.99	6.9949	7.0115	6.9985	-0.24	-0.05
4	3	10	1.20	10.0279	10.0435	10.0393	-0.16	-0.11
4	3	25	0.50	1.6582	1.6626	1.6572	-0.26	0.06
4	3	25	0.90	7.0152	7.0354	7.0113	-0.29	0.06
4	3	25	0.99	13.7693	13.7895	13.7720	-0.15	-0.02
4	3	25	1.20	24.6511	24.6669	24.6639	-0.06	-0.05
4	3	50	0.50	1.6582	1.6626	1.6572	-0.26	0.06
4	3	50	0.90	7.3288	7.3495	7.3240	-0.28	0.07
4	3	50	0.99	23.7064	23.7126	23.6919	-0.03	0.06
4	3	50	1.20	49.6410	49.6567	49.6538	-0.03	-0.03
4	3	100	0.50	1.6582	1.6626	1.6572	-0.26	0.06
4	3	100	0.90	7.3384	7.3591	7.3336	-0.28	0.07
4	3	100	0.99	38.8629	38.7713	38.7462	0.24	0.30
4	3	100	1.20	99.6410	99.6567	99.6538	-0.02	-0.01
4	5	5	0.50	2.5853	2.5890	2.5846	-0.14	0.03
4	5	5	0.90	5.5290	5.5453	5.5352	-0.29	-0.11
4	5	5	1.20	7.5636	7.5847	7.5801	-0.28	-0.22
4	5	15	0.50	2.5906	2.5944	2.5897	-0.15	0.03
4	5	15	0.90	7.7533	7.7769	7.7522	-0.30	0.01
4	5	15	0.99	11.0449	11.0699	11.0512	-0.23	-0.06
4	5	15	1.20	16.6764	16.7007	16.6949	-0.15	-0.11
4	5	50	0.50	2.5906	2.5944	2.5897	-0.15	0.03
4	5	50	0.90	8.8418	8.8689	8.8383	-0.31	0.04
4	5	50	0.99	25.4232	25.4559	25.4299	-0.13	-0.03
4	5	50	1.20	51.5606	51.5852	51.5798	-0.05	-0.04

Table 4.26: Expected Number of Customers in System, Part 4

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	10	5	0.50	5.0243	5.0264	5.0240	-0.04	0.00
4	10	5	0.90	9.6153	9.6356	9.6268	-0.21	-0.12
4	10	5	1.00	10.6921	10.7167	10.7103	-0.23	-0.17
4	10	5	1.20	12.2980	12.3285	12.3243	-0.25	-0.21
4	15	5	0.50	7.5076	7.5086	7.5075	-0.01	0.00
4	15	5	0.90	13.8252	13.8447	13.8397	-0.14	-0.10
4	15	5	1.20	17.1095	17.1440	17.1418	-0.20	-0.19
4	15	30	0.50	7.5087	7.5099	7.5086	-0.01	0.00
4	15	30	0.90	16.8527	16.9007	16.8494	-0.29	0.02
4	15	30	0.99	26.6507	26.7072	26.6636	-0.21	-0.05
4	15	30	1.20	41.3226	41.3704	41.3600	-0.12	-0.09
4	18	3	0.50	9.0010	9.0012	9.0011	-0.00	-0.00
4	18	3	0.90	15.6476	15.6496	15.6622	-0.01	-0.09
4	18	3	1.20	18.3204	18.3395	18.3474	-0.10	-0.15
4	18	10	0.50	9.0046	9.0053	9.0045	-0.01	0.00
4	18	10	0.90	17.7269	17.7628	17.7348	-0.20	-0.04
4	18	10	1.20	24.5484	24.5972	24.5882	-0.20	-0.16
4	50	25	0.50	N/A	25.0000	25.0000	N/A	N/A
4	50	25	0.90	N/A	47.0121	46.9487	N/A	N/A
4	50	25	1.20	N/A	71.0279	71.0233	N/A	N/A

Table 4.27: Expected Number of Customers in System, Part 5

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	3	1	0.50	1.4731	1.4729	1.4737	0.01	-0.04
5	3	1	0.90	2.3600	2.3605	2.3639	-0.02	-0.16
5	3	1	1.20	2.7821	2.7847	2.7886	-0.10	-0.23
5	3	10	0.50	1.6529	1.6563	1.6516	-0.21	0.08
5	3	10	0.90	5.3140	5.3275	5.3134	-0.26	0.01
5	3	10	0.99	7.0008	7.0154	7.0055	-0.21	-0.07
5	3	10	1.20	10.0877	10.1036	10.1026	-0.16	-0.15
5	3	25	0.50	1.6530	1.6564	1.6516	-0.21	0.08
5	3	25	0.90	6.8838	6.8993	6.8788	-0.23	0.07
5	3	25	0.99	13.7483	13.7656	13.7520	-0.13	-0.03
5	3	25	1.20	24.7565	24.7729	24.7731	-0.07	-0.07
5	3	50	0.50	1.6530	1.6564	1.6516	-0.21	0.08
5	3	50	0.90	7.1502	7.1658	7.1442	-0.22	0.08
5	3	50	0.99	23.5873	23.5859	23.5693	0.01	0.08
5	3	50	1.20	49.7492	49.7655	49.7658	-0.03	-0.03
5	3	100	0.50	1.6530	1.6564	1.6516	-0.21	0.08
5	3	100	0.90	7.1570	7.1725	7.1508	-0.22	0.09
5	3	100	0.99	38.3732	38.2639	38.2432	0.28	0.34
5	3	100	1.20	99.7492	99.7655	99.7657	-0.02	-0.02

Table 4.28: Expected Number of Customers in System, Part 6

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	5	5	0.50	2.5832	2.5864	2.5824	-0.12	0.03
5	5	5	0.90	5.5310	5.5479	5.5391	-0.31	-0.15
5	5	5	1.20	7.5874	7.6116	7.6091	-0.32	-0.29
5	5	15	0.50	2.5879	2.5912	2.5869	-0.13	0.04
5	5	15	0.90	7.6947	7.7162	7.6933	-0.28	0.02
5	5	15	0.99	11.0499	11.0744	11.0581	-0.22	-0.07
5	5	15	1.20	16.7584	16.7852	16.7826	-0.16	-0.14
5	5	50	0.50	2.5879	2.5912	2.5869	-0.13	0.04
5	5	50	0.90	8.6765	8.7004	8.6718	-0.27	0.05
5	5	50	0.99	25.3108	25.3427	25.3195	-0.13	-0.03
5	5	50	1.20	51.6627	51.6898	51.6876	-0.05	-0.05
5	10	5	0.50	5.0239	5.0257	5.0235	-0.04	0.01
5	10	5	0.90	9.6214	9.6439	9.6365	-0.23	-0.16
5	10	5	1.00	10.7064	10.7350	10.7305	-0.27	-0.23
5	10	5	1.20	12.3206	12.3569	12.3554	-0.30	-0.28
5	15	5	0.50	7.5075	7.5084	7.5074	-0.01	0.00
5	15	5	0.90	13.8330	13.8561	13.8522	-0.17	-0.14
5	15	5	1.20	17.1309	17.1735	17.1737	-0.25	-0.25
5	15	30	0.50	7.5086	7.5095	7.5084	-0.01	0.00
5	15	30	0.90	16.7469	16.7919	16.7436	-0.27	0.02
5	15	30	0.99	26.6397	26.6957	26.6567	-0.21	-0.06
5	15	30	1.20	41.4055	41.4600	41.4544	-0.13	-0.12

Table 4.29: Expected Number of Customers in System, Part 7

k	n	q	ρ	Number of Customers			% Difference	
				Exact	ELP	ELC	ELP	ELC
8	3	10	0.50	1.6450	1.6470	1.6432	-0.12	0.11
8	3	10	0.90	5.2647	5.2757	5.2639	-0.21	0.02
8	3	10	1.20	10.1822	10.2022	10.2033	-0.20	-0.21
8	5	10	0.50	2.5839	2.5858	2.5825	-0.08	0.05
8	5	10	0.90	6.8399	6.8566	6.8422	-0.24	-0.03
8	5	10	1.20	12.0710	12.1033	12.1035	-0.27	-0.27
10	3	2	0.50	1.5664	1.5684	1.5672	-0.13	-0.05
10	3	2	0.90	2.8161	2.8272	2.8274	-0.39	-0.40
10	3	2	1.20	3.4836	3.5008	3.5025	-0.49	-0.54
10	3	10	0.50	1.6423	1.6437	1.6404	-0.08	0.11
10	3	10	0.90	5.2467	5.2562	5.2459	-0.18	0.02
10	3	10	1.20	10.2151	10.2371	10.2385	-0.22	-0.23
10	5	10	0.50	2.5825	2.5838	2.5810	-0.05	0.06
10	5	10	0.90	6.8254	6.8403	6.8280	-0.22	-0.04
10	5	10	1.20	12.1022	12.1374	12.1383	-0.29	-0.30
10	10	5	0.50	5.0230	5.0239	5.0225	-0.02	0.01
10	10	5	0.90	9.6327	9.6611	9.6569	-0.30	-0.25
10	10	5	1.20	12.3683	12.4219	12.4237	-0.43	-0.45
15	3	10	0.50	1.6387	1.6390	1.6367	-0.02	0.12
15	3	10	0.90	5.2213	5.2279	5.2205	-0.13	0.02
15	3	10	1.20	10.2602	10.2852	10.2868	-0.24	-0.26
15	5	10	0.50	2.5807	2.5810	2.5791	-0.01	0.06
15	5	10	0.90	6.8047	6.8167	6.8080	-0.18	-0.05
15	5	10	1.20	12.1450	12.1847	12.1860	-0.33	-0.34
20	3	10	0.50	1.6368	1.6366	1.6348	0.01	0.13
20	3	10	0.90	5.2078	5.2131	5.2072	-0.10	0.01
20	3	10	1.20	10.2834	10.3100	10.3116	-0.26	-0.27

Kolmogorov equations needed to solve the models and the CPU times required to obtain the solutions.

To illustrate the advantage of using the heuristic techniques, we compare in Tables 4.37 and 4.38 the number of Chapman-Kolmogorov equations and the associated CPU times required to solve these equations for each of the 170 models in the study. The first three columns specify all combinations of n, q and ρ examined for each value of k. Columns 4 and 5 specify the number of states in each model for the exact and heuristic techniques, respectively. The CPU times required to obtain steady-state solutions with $\rho = 0.9$ for the exact solution technique is in column 6, and the maximum of the times required using ELC and ELP is in column 7. The CPU times required to solve the models using the two heuristics are almost identical. In some cases, it takes longer to solve the model using ELP than using ELC, and vice versa, but the differences are small. The final column shows the ratios of columns 6 and 7. We selected the utilization ratio of 0.9 because it is a high utilization ratio without being over-saturated. As seen in Table 4.1, it takes longer to reach steady-state for models with $\rho = 0.9$ than with $\rho = 0.5$ or 1.2, due to longer transient periods.

The maximum number of states using the exact solution technique grows faster, as suggested by Equation 3.1, than the number of states using the heuristic solution techniques, as given in Equation 3.18. For example, the ranges for the models presented go from 30 states up to 901901 states in the case of the exact solution, while for the same models using the heuristics the range is from 23 to 7651. This number is particularly sensitive to the value of k.

The CPU times for running the experiments using the exact solution technique range from 0.42 seconds up to 192,475 seconds (53.46 hours); note that we were unable to solve the largest system using the exact solution technique with the available computer hardware and software. Using the heuristic solution techniques, solution times ranged from 0.25 seconds up to 1183.73 seconds (19.73 minutes). The time to obtain the solution not only depends on the number of Chapman-Kolmogorov equations to solve, but also on how long the model takes to reach steady-state. For example, solving the $M/E_4/5/5+15$ system, with 966 states, using the exact solution

Table 4.30: Expected Number of Busy Servers, Part 1

k	n	q	ρ	Number of Busy Servers			% Difference	
				Exact	ELP	ELC	ELP	ELC
3	3	1	0.50	1.4169	1.4157	1.4176	0.09	-0.05
3	3	1	0.90	2.1507	2.1496	2.1534	0.05	-0.12
3	3	1	1.20	2.4558	2.4565	2.4597	-0.03	-0.16
3	3	10	0.50	1.5000	1.5000	1.5000	0.00	0.00
3	3	10	0.90	2.6426	2.6425	2.6427	0.00	-0.00
3	3	10	0.99	2.8136	2.8135	2.8137	0.00	-0.01
3	3	10	1.20	2.9750	2.9751	2.9751	-0.00	-0.00
3	3	25	0.50	1.5000	1.5000	1.5000	0.00	0.00
3	3	25	0.90	2.6953	2.6953	2.6953	-0.00	-0.00
3	3	25	0.99	2.9092	2.9092	2.9093	0.00	-0.00
3	3	25	1.20	2.9996	2.9996	2.9996	-0.00	-0.00
3	3	50	0.50	1.5000	1.5000	1.5000	0.00	0.00
3	3	50	0.90	2.6999	2.6998	2.7000	0.00	-0.00
3	3	50	0.99	2.9445	2.9444	2.9445	0.00	0.00
3	3	50	1.20	3.0000	3.0000	3.0000	0.00	0.00
3	3	100	0.50	1.5000	1.5000	1.5000	0.00	0.00
3	3	100	0.90	2.7001	2.6999	2.7000	0.01	0.00
3	3	100	0.99	2.9616	2.9614	2.9614	0.00	0.00
3	3	100	1.20	3.0000	3.0000	3.0000	0.00	0.00
3	5	5	0.50	2.4981	2.4980	2.4981	0.00	-0.00
3	5	5	0.90	4.2493	4.2490	4.2503	0.01	-0.02
3	5	5	1.20	4.8249	4.8252	4.8257	-0.01	-0.02
3	5	15	0.50	2.5001	2.5000	2.5001	0.00	-0.00
3	5	15	0.90	4.4625	4.4624	4.4626	0.00	-0.00
3	5	15	0.99	4.7781	4.7781	4.7783	0.00	-0.00
3	5	15	1.20	4.9906	4.9906	4.9907	-0.00	-0.00
3	5	50	0.50	2.5001	2.5000	2.5001	0.00	-0.00
3	5	50	0.90	4.4998	4.4999	4.4999	-0.00	-0.00
3	5	50	0.99	4.9079	4.9079	4.9079	-0.00	-0.00
3	5	50	1.20	5.0000	5.0000	5.0000	0.00	0.00

Table 4.31: Expected Number of Busy Servers, Part 2

k	n	q	ρ	Number of Busy Servers			% Difference	
				Exact	ELP	ELC	ELP	ELC
3	10	5	0.50	4.9986	4.9986	4.9987	0.00	-0.00
3	10	5	0.90	8.5675	8.5669	8.5701	0.01	-0.03
3	10	5	1.00	9.1062	9.1069	9.1093	-0.01	-0.03
3	10	5	1.20	9.6970	9.6999	9.6996	-0.03	-0.03
3	15	5	0.50	7.4993	7.4992	7.4993	0.00	-0.00
3	15	5	0.90	12.9236	12.9230	12.9279	0.00	-0.03
3	15	5	1.20	14.5915	14.5984	14.5963	-0.05	-0.03
3	15	30	0.50	7.4997	7.4999	7.4999	-0.00	-0.00
3	15	30	0.90	13.4915	13.4912	13.4916	0.00	-0.00
3	15	30	0.99	14.6165	14.6167	14.6168	-0.00	-0.00
3	15	30	1.20	14.9994	14.9996	15.0002	-0.00	-0.01
3	18	3	0.50	8.9974	8.9971	8.9975	0.00	-0.00
3	18	3	0.90	15.2172	15.2106	15.2240	0.04	-0.05
3	18	3	1.20	17.1519	17.1609	17.1618	-0.05	-0.06
3	18	10	0.50	8.9997	9.0001	8.9998	-0.00	-0.00
3	18	10	0.90	15.9471	15.9474	15.9486	-0.00	-0.01
3	18	10	1.20	17.8927	17.8949	17.8935	-0.01	-0.00
3	50	25	0.50	24.9990	24.9996	25.0000	-0.00	-0.00
3	50	25	0.90	44.9572	44.9579	44.9577	-0.00	-0.00
3	50	25	1.20	49.9965	49.9968	49.9965	-0.00	-0.00

Table 4.32: Expected Number of Busy Servers, Part 3

k	n	q	ρ	Number of Busy Servers Exact	ELP	ELC	% Difference ELP	ELC
4	3	1	0.50	1.4185	1.4173	1.4197	0.08	-0.09
4	3	1	0.90	2.1564	2.1554	2.1611	0.04	-0.22
4	3	1	1.20	2.4630	2.4643	2.4698	-0.06	-0.28
4	3	10	0.50	1.5000	1.5000	1.4999	0.00	0.01
4	3	10	0.90	2.6485	2.6484	2.6487	0.00	-0.01
4	3	10	0.99	2.8220	2.8219	2.8223	0.00	-0.01
4	3	10	1.20	2.9793	2.9793	2.9794	0.00	-0.00
4	3	25	0.50	1.5000	1.5000	1.4999	0.00	0.00
4	3	25	0.90	2.6964	2.6963	2.6964	0.00	-0.00
4	3	25	0.99	2.9134	2.9134	2.9135	0.00	-0.00
4	3	25	1.20	2.9998	2.9998	2.9998	0.00	0.00
4	3	50	0.50	1.5000	1.5000	1.4999	0.00	0.00
4	3	50	0.90	2.7000	2.6998	2.6999	0.01	0.00
4	3	50	0.99	2.9467	2.9466	2.9466	0.00	0.00
4	3	50	1.20	3.0000	3.0000	3.0000	0.00	0.00
4	3	100	0.50	1.5000	1.5000	1.4999	0.00	0.00
4	3	100	0.90	2.6999	2.6998	2.7000	0.01	-0.00
4	3	100	0.99	2.9625	2.9623	2.9623	0.01	0.01
4	3	100	1.20	3.0000	3.0000	3.0000	0.00	0.00
4	5	5	0.50	2.4984	2.4983	2.4984	0.00	-0.00
4	5	5	0.90	4.2630	4.2625	4.2646	0.01	-0.04
4	5	5	1.20	4.8396	4.8398	4.8410	-0.00	-0.03
4	5	15	0.50	2.5000	2.4999	2.4999	0.00	0.00
4	5	15	0.90	4.4677	4.4676	4.4678	0.00	-0.00
4	5	15	0.99	4.7881	4.7880	4.7884	0.00	-0.01
4	5	15	1.20	4.9929	4.9929	4.9929	-0.00	-0.00
4	5	50	0.50	2.5000	2.4999	2.4999	0.00	0.00
4	5	50	0.90	4.4999	4.5000	4.5001	-0.00	-0.00
4	5	50	0.99	4.9114	4.9114	4.9115	-0.00	-0.00
4	5	50	1.20	5.0000	5.0000	5.0000	0.00	0.00

Table 4.33: Expected Number of Busy Servers, Part 4

k	n	q	ρ	Number of Busy Servers			% Difference	
				Exact	ELP	ELC	ELP	ELC
4	10	5	0.50	4.9988	4.9987	4.9989	0.00	-0.00
4	10	5	0.90	8.5859	8.5847	8.5904	0.01	-0.05
4	10	5	1.00	9.1300	9.1298	9.1353	0.00	-0.06
4	10	5	1.20	9.7178	9.7195	9.7221	-0.02	-0.04
4	15	5	0.50	7.4993	7.4993	7.4994	0.00	-0.00
4	15	5	0.90	12.9436	12.9413	12.9512	0.02	-0.06
4	15	5	1.20	14.6146	14.6189	14.6228	-0.03	-0.06
4	15	30	0.50	7.4998	7.5001	7.4999	-0.00	-0.00
4	15	30	0.90	13.4936	13.4936	13.4937	0.00	-0.00
4	15	30	0.99	14.6322	14.6320	14.6326	0.00	-0.00
4	15	30	1.20	14.9998	14.9998	14.9998	0.00	0.00
4	18	3	0.50	8.9975	8.9972	8.9976	0.00	-0.00
4	18	3	0.90	15.2306	15.2227	15.2427	0.05	-0.08
4	18	3	1.20	17.1688	17.1759	17.1865	-0.04	-0.10
4	18	10	0.50	8.9999	8.9998	8.9998	0.00	0.00
4	18	10	0.90	15.9672	15.9654	15.9696	0.01	-0.02
4	18	10	1.20	17.9076	17.9080	17.9089	-0.00	-0.01
4	50	25	0.50	N/A	25.0000	25.0000	N/A	N/A
4	50	25	0.90	N/A	44.9644	44.9657	N/A	N/A
4	50	25	1.20	N/A	49.9977	49.9977	N/A	N/A

Table 4.34: Expected Number of Busy Servers, Part 5

k	n	q	ρ	Number of Busy Servers			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	3	1	0.50	1.4194	1.4188	1.4211	0.04	-0.11
5	3	1	0.90	2.1600	2.1607	2.1661	-0.03	-0.28
5	3	1	1.20	2.4675	2.4711	2.4765	-0.15	-0.37
5	3	10	0.50	1.5000	1.5000	1.4999	-0.00	0.00
5	3	10	0.90	2.6521	2.6521	2.6523	0.00	-0.01
5	3	10	0.99	2.8272	2.8272	2.8276	0.00	-0.01
5	3	10	1.20	2.9817	2.9817	2.9819	-0.00	-0.00
5	3	25	0.50	1.5000	1.5000	1.4999	-0.00	0.00
5	3	25	0.90	2.6968	2.6969	2.6969	-0.00	-0.00
5	3	25	0.99	2.9160	2.9160	2.9160	-0.00	-0.00
5	3	25	1.20	2.9998	2.9998	2.9998	0.00	0.00
5	3	50	0.50	1.5000	1.5000	1.4999	-0.00	0.00
5	3	50	0.90	2.7000	2.7000	2.6999	0.00	0.00
5	3	50	0.99	2.9480	2.9479	2.9479	0.00	0.00
5	3	50	1.20	3.0000	3.0000	3.0000	0.00	0.00
5	3	100	0.50	1.5000	1.5000	1.4999	-0.00	0.00
5	3	100	0.90	2.6999	2.6999	2.6998	0.00	0.00
5	3	100	0.99	2.9630	2.9629	2.9629	0.00	0.00
5	3	100	1.20	3.0000	3.0000	3.0000	0.00	0.00

Table 4.35: Expected Number of Busy Servers, Part 6

k	n	q	ρ	Number of Busy Servers			% Difference	
				Exact	ELP	ELC	ELP	ELC
5	5	5	0.50	2.4985	2.4985	2.4986	0.00	-0.00
5	5	5	0.90	4.2716	4.2715	4.2736	0.00	-0.05
5	5	5	1.20	4.8488	4.8491	4.8505	-0.01	-0.04
5	5	15	0.50	2.4999	2.4999	2.5000	-0.00	-0.01
5	5	15	0.90	4.4707	4.4707	4.4709	0.00	-0.00
5	5	15	0.99	4.7942	4.7942	4.7946	0.00	-0.01
5	5	15	1.20	4.9941	4.9941	4.9941	-0.00	-0.00
5	5	50	0.50	2.4999	2.4999	2.5000	-0.00	-0.01
5	5	50	0.90	4.4999	4.4999	4.4999	0.00	0.00
5	5	50	0.99	4.9135	4.9136	4.9136	-0.00	-0.00
5	5	50	1.20	5.0000	5.0000	5.0000	0.00	0.00
5	10	5	0.50	4.9989	4.9989	4.9990	0.00	-0.00
5	10	5	0.90	8.5977	8.5975	8.6034	0.00	-0.07
5	10	5	1.00	9.1451	9.1462	9.1520	-0.01	-0.07
5	10	5	1.20	9.7308	9.7333	9.7362	-0.03	-0.06
5	15	5	0.50	7.4994	7.4993	7.4994	0.00	-0.00
5	15	5	0.90	12.9563	12.9563	12.9661	0.00	-0.08
5	15	5	1.20	14.6291	14.6353	14.6396	-0.04	-0.07
5	15	30	0.50	7.4999	7.5000	7.4998	-0.00	0.00
5	15	30	0.90	13.4946	13.4947	13.4948	-0.00	-0.00
5	15	30	0.99	14.6417	14.6416	14.6422	0.00	-0.00
5	15	30	1.20	14.9999	14.9999	14.9999	0.00	0.00

Table 4.36: Expected Number of Busy Servers, Part 7

k	n	q	ρ	Number of Busy Servers Exact	ELP	ELC	% Difference ELP	ELC
8	3	10	0.50	1.5000	1.5000	1.5000	0.00	-0.00
8	3	10	0.90	2.6574	2.6575	2.6577	-0.00	-0.01
8	3	10	1.20	2.9851	2.9852	2.9853	-0.00	-0.01
8	5	10	0.50	2.5000	2.5000	2.5000	-0.00	-0.00
8	5	10	0.90	4.4322	4.4323	4.4328	-0.00	-0.01
8	5	10	1.20	4.9765	4.9767	4.9769	-0.00	-0.01
10	3	2	0.50	1.4721	1.4728	1.4735	-0.04	-0.09
10	3	2	0.90	2.3494	2.3525	2.3554	-0.13	-0.26
10	3	2	1.20	2.6911	2.6956	2.6985	-0.17	-0.27
10	3	10	0.50	1.5000	1.5000	1.5000	0.00	0.00
10	3	10	0.90	2.6592	2.6593	2.6595	-0.00	-0.01
10	3	10	1.20	2.9862	2.9863	2.9864	-0.00	-0.01
10	5	10	0.50	2.5000	2.5001	2.5000	-0.00	0.00
10	5	10	0.90	4.4349	4.4351	4.4355	-0.01	-0.01
10	5	10	1.20	4.9782	4.9784	4.9785	-0.00	-0.01
10	10	5	0.50	4.9990	4.9991	4.9992	-0.00	-0.00
10	10	5	0.90	8.6225	8.6268	8.6310	-0.05	-0.10
10	10	5	1.20	9.7586	9.7636	9.7660	-0.05	-0.08
15	3	10	0.50	1.5000	1.5000	1.5000	0.00	0.00
15	3	10	0.90	2.6615	2.6617	2.6619	-0.01	-0.01
15	3	10	1.20	2.9876	2.9877	2.9878	-0.00	-0.01
15	5	10	0.50	2.5000	2.5000	2.4999	-0.00	0.00
15	5	10	0.90	4.4385	4.4389	4.4392	-0.01	-0.02
15	5	10	1.20	4.9803	4.9806	4.9807	-0.00	-0.01
20	3	10	0.50	1.5000	1.5001	1.5000	-0.00	-0.00
20	3	10	0.90	2.6627	2.6629	2.6630	-0.01	-0.01
20	3	10	1.20	2.9883	2.9884	2.9885	-0.00	-0.01

technique takes 140.96 seconds, and solving the $M/E_5/3/3+25$ system, with 931 states, also using the exact approach takes 256.01 seconds. Using the heuristic solution techniques, the same effect occurs; for example, let us compare the time to solve the $M/E_5/15/15+5$ queue, with 801 states, with the time to solve the $M/E_4/5/5+50$ queue, with 851 states. The former takes 44.87 seconds while the latter takes 141.55 seconds.

As a final comment on Tables 4.37 and 4.38, the CPU time ratios illustrate the clear advantage of using the heuristic solution techniques over the exact solution technique. The importance in the speed of solving the desired systems resides in the use of the technique for solving multiple queueing systems or a network of queues. If the heuristic techniques were to require longer times, modeling the system would not be useful in applications. For example, the application in Section 5.2 could be included in a network of multiple en-route sectors, each modeled by an $M(t)/E_k(t)/n/n+q$ queueing system. Notice the range of the CPU time ratios: from 1.68 times faster in the smallest system up to 2646.07 times for the second largest system considered in the analysis (the largest we were able to solve using both exact and heuristic solution techniques).

The computing times for both heuristics are similar and always in the same order of magnitude. Therefore, for the remainder of the thesis, ELC will be used as the heuristic of choice.

In this section, we have presented an extensive collection of examples that validate the use of ELC to approximate steady-state behavior of $M/E_k/n/n+q$ systems under stationary conditions. The examples described above include a wide range of parameters k, n, q and ρ. The results presented for both occupancy probabilities and several aggregate performance measures, along with information on the system size (number of states) and CPU times, show strong evidence that ELC is an excellent alternative to the exact solution technique: 95% of the ELC results are within 1% of the exact values, and 100% of the results are within 3%; systems that we were unable to solve using the exact solution technique, because of the large number of states, can be solved using ELC; and, the CPU times to solve the models using ELC are much faster than using the exact model, up to three orders of magnitude faster. The heuristic solution technique ELC is very robust to parameter changes and its excellent performance appears to be quite robust to parameter changes.

4.2 Validation: Transient Conditions

In this section, we present behavior of several models during their initial epochs showing their response to initial conditions. We continue our validation of the ELC heuristic by comparing transient results generated by the exact and ELC solution techniques. As with the steady-state, the ELC heuristic provides and excellent approximation to transient model response.

The examples below are systems with finite queue size and stationary parameters. Some examples have large enough waiting room that they behave as systems with (effectively) infinite waiting capacity. All systems start empty and idle. The performance measures examined are the expected number of customers in the system and expected virtual delay, as defined in Section 3.2. The reason for choosing such performance measures is to analyze how fast the system becomes busy and how long it takes for the delays to become considerable, given the arrival and service rates of the queue. In all examples, the service rate is held constant to $\mu = 0.1$, which corresponds to the same service rate as in Section 4.1 of 0.1 customers per epoch per server. The epochs in this section are smaller than in Section 4.1. In this section, the unit of time is one minute, compared to one hour in Section 4.1 (e.g., we are looking the system every minute instead of every hour).

We tested six different models with various utilization ratios for a total of twelve examples. The examples cover a wide range of system parameters for the Erlang order k, the number of servers n, and the size of the queue q. We used three different utilization ratios, $\rho = 0.5$, 0.9 and 1.2, as in the previous section (see Table 4.39).

Table 4.39 shows the maximum percentage difference between the exact and ELC results, for each example, in columns 3 and 4, and the actual differences in columns 5 and 6. Notice that the maximum percentage differences were up to 14.46% while

Table 4.37: Summary of Examples, Number of States and CPU Times for Erlang Orders 3 and 4

Servers n	Queue q	Utilization ρ	Number of States Exact	Number of States Heuristic	CPU Times (sec), $\rho = 0.9$ Exact	CPU Times (sec), $\rho = 0.9$ Heuristic (max)	CPU Times (sec), $\rho = 0.9$ Ratio
Erlang Order (k): 3							
3	1	0.5, 0.9, 1.2	30	23	0.42	0.25	1.68
3	10	0.5, 0.9, 0.99, 1.2	120	86	10.16	1.71	5.94
3	25	0.5, 0.9, 0.99, 1.2	270	191	38.14	10.76	3.54
3	50	0.5, 0.9, 0.99, 1.2	520	366	136.47	35.56	3.84
3	100	0.5, 0.9, 0.99, 1.2	1020	716	373.73	87.66	4.26
5	5	0.5, 0.9, 1.2	161	91	5.32	1.14	4.67
5	15	0.5, 0.9, 0.99, 1.2	371	201	39.05	6.28	6.22
5	50	0.5, 0.9, 0.99, 1.2	1106	586	227.75	74.08	3.07
10	5	0.5, 0.9, 0.99, 1.2	616	226	34.85	10.03	3.47
15	5	0.5, 0.9, 1.2	1496	411	98.55	14.89	6.62
15	30	0.5, 0.9, 0.99, 1.2	4896	1186	1155.5	122.76	9.41
18	3	0.5, 0.9, 1.2	1900	472	156.13	20.05	7.79
18	10	0.5, 0.9, 1.2	3230	731	574.52	64.03	8.97
50	25	0.5, 0.9, 1.2	56576	5126	13221.5	661.53	19.99
Erlang Order (k): 4							
3	1	0.5, 0.9, 1.2	55	32	0.96	0.33	2.91
3	10	0.5, 0.9, 0.99, 1.2	235	122	15.49	5.12	3.03
3	25	0.5, 0.9, 0.99, 1.2	535	272	101.46	18.21	5.57
3	50	0.5, 0.9, 0.99, 1.2	1035	522	369.77	61.33	6.03
3	100	0.5, 0.9, 0.99, 1.2	2035	1022	844.52	147.82	5.71
5	5	0.5, 0.9, 1.2	406	131	18.14	2.05	8.85
5	15	0.5, 0.9, 0.99, 1.2	966	291	140.96	10.96	12.86
5	50	0.5, 0.9, 0.99, 1.2	2926	851	825.23	141.55	5.83
10	5	0.5, 0.9, 0.99, 1.2	2431	331	198.43	18.66	10.63
15	5	0.5, 0.9, 1.2	7956	606	1022.69	26.25	38.96
15	30	0.5, 0.9, 0.99, 1.2	28356	1756	10132.98	200.45	50.55
18	3	0.5, 0.9, 1.2	11305	697	2270.3	63.46	35.78
18	10	0.5, 0.9, 1.2	20615	1082	6020.9	154.95	38.86
50	25	0.5, 0.9, 1.2	901901	7651	N/A	1183.73	N/A

Table 4.38: Summary of Examples, Number of States and CPU Times for Erlang Orders 5, 8, 10, 15 and 20

Servers n	Queue q	Utilization ρ	Number of States Exact	Number of States Heuristic	CPU Times (sec), $\rho = 0.9$ Exact	CPU Times (sec), $\rho = 0.9$ Heuristic (max)	CPU Times (sec), $\rho = 0.9$ Ratio
Erlang Order (k): 5							
3	1	0.5, 0.9, 1.2	91	41	4.17	0.84	4.96
3	10	0.5, 0.9, 0.99, 1.2	406	158	38.54	4.68	8.24
3	25	0.5, 0.9, 0.99, 1.2	931	353	256.01	31.1	8.23
3	50	0.5, 0.9, 0.99, 1.2	1806	678	1286.24	99.27	12.96
3	100	0.5, 0.9, 0.99, 1.2	3556	1328	2153.3	207.43	10.38
5	5	0.5, 0.9, 1.2	882	171	55.77	3.01	18.53
5	15	0.5, 0.9, 0.99, 1.2	2142	381	453.7	18.14	25.01
5	50	0.5, 0.9, 0.99, 1.2	6552	1116	2789.7	250.93	11.12
10	5	0.5, 0.9, 0.99, 1.2	8008	436	939.26	26.35	35.65
15	5	0.5, 0.9, 1.2	34884	801	5048.7	44.87	112.52
15	30	0.5, 0.9, 0.99, 1.2	131784	2326	55031.1	425.29	129.40
Erlang Order (k): 8							
3	10	0.5, 0.9, 1.2	1365	266	266.62	14.64	18.21
5	10	0.5, 0.9, 1.2	9207	471	3704.2	22.48	164.78
Erlang Order (k): 10							
3	2	0.5, 0.9, 1.2	726	114	47.45	2.2	21.57
3	10	0.5, 0.9, 1.2	2486	338	692.66	22.76	30.43
5	10	0.5, 0.9, 1.2	23023	601	13125.1	34.54	380.00
10	5	0.5, 0.9, 1.2	646646	961	192475.05	72.74	2646.07
Erlang Order (k): 15							
3	10	0.5, 0.9, 1.2	7616	518	4121.3	49.73	82.87
5	10	0.5, 0.9, 1.2	131784	926	78712.5	78.33	1004.88
Erlang Order (k): 20							
3	10	0.5, 0.9, 1.2	17171	698	25712.4	88.33	291.09

Table 4.39: Transient Examples and Maximum Percentage Difference

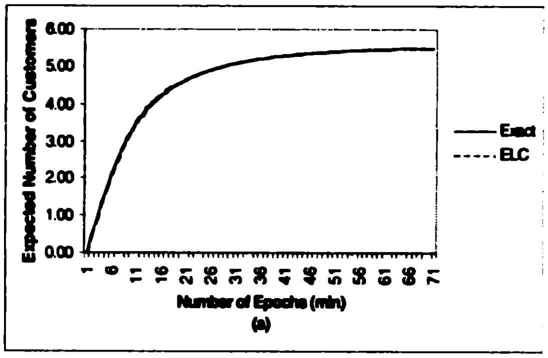
System	Utilization ratio ρ	% Difference E[Cust.]	% Difference E[Delay]	Actual Difference E[Cust.]	Actual Difference E[Delay] (minutes)
$M/E_3/5/5+5$	0.5	0.36	2.6	0.0065	0.0021
	0.9	0.48	2.18	0.0147	0.0107
	1.2	0.54	1.92	0.0220	0.0239
$M/E_3/18/18+3$	0.5	0.54	5.88	0.0332	0.0001
	0.9	0.58	6.57	0.0575	0.0003
	1.2	0.57	5.06	0.0760	0.0031
$M/E_4/15/15+30$	0.5	1.07	14.46	0.0561	0.0001
	0.9	1.16	10.50	0.1101	0.0041
$M/E_5/3/3+100$	0.9	1.08	3.01	0.0241	0.0701
$M/E_{10}/3/3+10$	0.9	2.36	6.19	0.0559	0.1612
	1.2	2.58	5.49	0.0818	0.2541
$M/E_{20}/3/3+10$	0.9	3.81	9.52	0.0939	0.2678

the actual differences were very small in all cases..

It is important to remember that we are not analyzing the behavior of the $M(t)/E_k(t)/n/n+q$ queueing system. Rather, we are interested in the comparison of the exact and ELC results to demonstrate that the heuristic solution technique approximates well transient behavior.

In order to better appreciate the behavior of the heuristic solution technique during the transient period, we show the plots of some of the examples in Table 4.39. We have selected three examples that are representative of the set of models analyzed, and we also present the three cases with the worst performance of ELC in approximating the exact results. Notice that the worst performances are observed in the transients for the expected virtual delay. The results for the expected number of customers are consistently good.

The plots in the examples below are not all drawn in the same axis scales, even for the results of the same model, as the transients are different for each performance



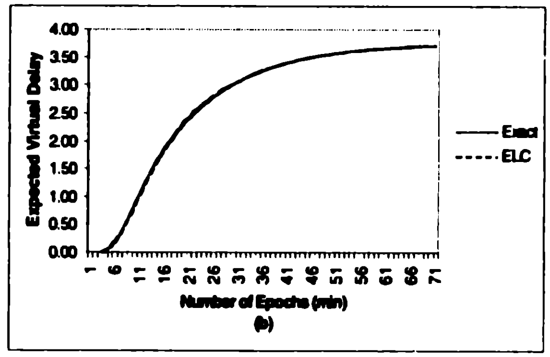


Figure 4-5: Transient results for the $M/E_3/5/5+5$ queueing system with $\rho = 0.9$: (a) L and (b) $E[Delay]$ (minutes)

measure and we want to analyze each measure separately. The first set of examples are for the $M/E_3/5/5+5$ queueing system. Figure 4-5 shows the transient results for the exact and ELC solution techniques, with $\rho = 0.9$. This small and very capacitated system (small k, n and q) demonstrates that the two solution techniques show almost identical results.

We increase the number of servers and reduce the queue capacity to obtain the $M/E_3/18/18+3$ queueing system. The transient results, for both exact and heuristic solution technique with $\rho = 0.9$ and 1.2, are presented in Figures 4-6. The expected number of customers is approximated extremely well with both utilization ratios, while the expected virtual delay shows slight differences between the results of both techniques. Figure 4-6, parts (b) and (d), are two of the examples with the worst performance we observed in approximating the exact results. Even though some differences between both results are evident, the ELC results still approximate very well the exact results and are always within 7% and 6% for $\rho = 0.9$ and 1.2, respectively, of the exact values.

Our next set of examples are the transients for the $M/E_4/15/15+30$ model. Figure 4-7 shows the results for the model with an utilization ratio of 0.5. Similarly as in the previous examples, the expected number of customers is clearly better than the expected virtual delay. The expected virtual delay in this example, illustrated in Figure 4-7, part (b), is the worst case of all the models in this section.

The last two sets of examples include the systems with extreme values of q and k.

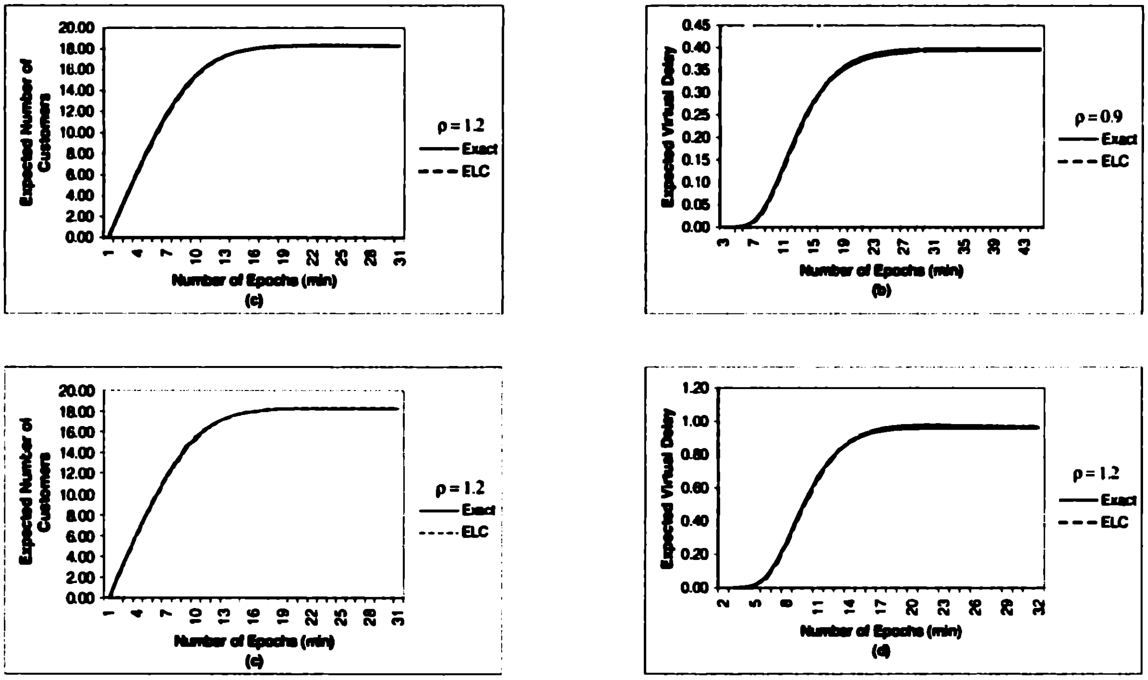


Figure 4-6: Transient results for the $M/E_3/18/18+3$ queueing system with $\rho = 0.9$ and 1.2: (a) and (c) L, and (b) and (d) $E[Delay]$ (minutes)

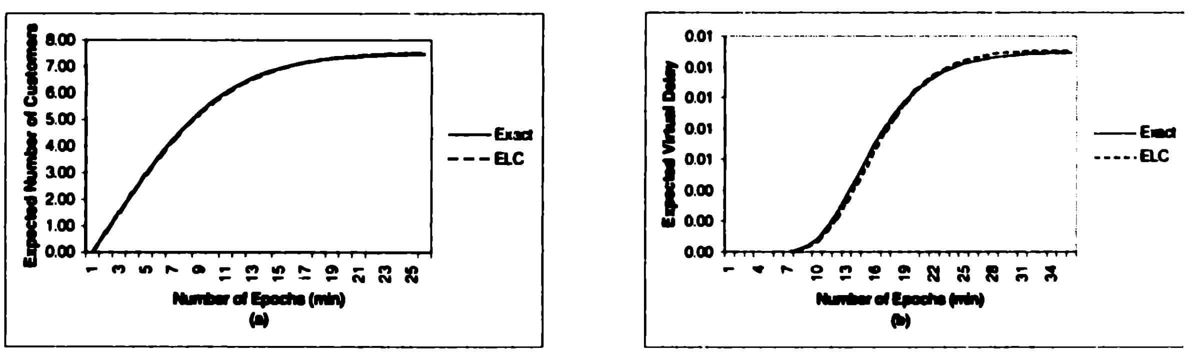


Figure 4-7: Transient results for the $M/E_4/15/15+30$ queueing system with $\rho = 0.5$: (a) Expected number of customers and (b) expected virtual delay (minutes)

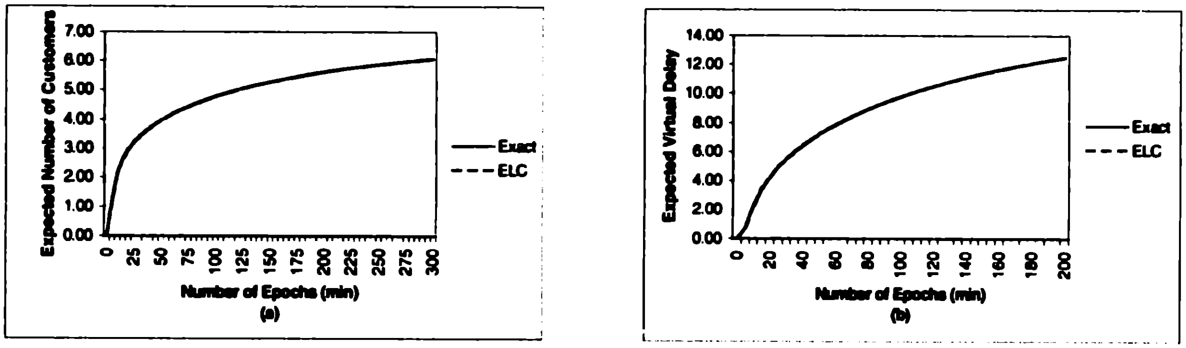


Figure 4-8: Transient results for the $M/E_5/3/3+100$ queueing system with $\rho = 0.9$: (a) L and (b) $E[Delay]$ (minutes)

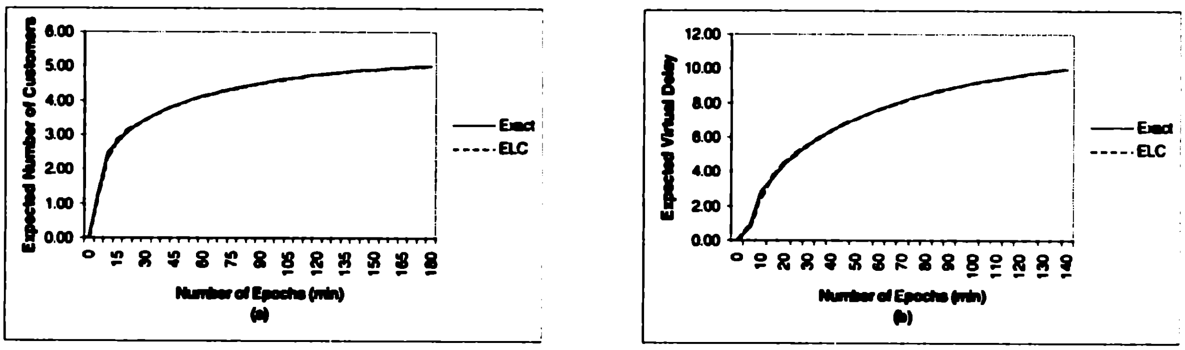


Figure 4-9: Transient results for the $M/E_{20}/3/3+10$ queueing system with $\rho = 0.9$: (a) L and (b) $E[Delay]$ (minutes)

Figures 4-8 and 4-9 show the results for the $M/E_5/3/3+100$ and $M/E_{20}/3/3+10$ queueing systems, respectively. In those two sets of examples, the results for both the expected number of customers and the expected virtual delay, are practically the same for the exact and heuristic solution techniques. The rest of the examples in Table 4.39 behave similarly in the sense that the results for the exact and ELC are almost indistinguishable from each other.

These results indicate that the heuristic proposed in Chapter 3 provides a very good approximation of behavior of $M(t)/E_k(t)/n/n+q$ systems even during the transient period from rest until the system reaches steady-state. The importance of analyzing the transient period of the $M(t)/E(t)/n/n+q$ queueing systems is because in most (if not all) applications, it is necessary to deal with starting conditions until we reach a "normal" stage of operation. Therefore, our heuristic may very well be used to analyze the real-life problems that can be modeled with multi-server systems with Poisson arrivals and Erlangian service time distributions.

In the queueing literature there are few authors that have addressed the transient analysis of multi-server systems. In Chapter 2, we described the work of Odoni and Roth [37] and Murray and Kelton [30] among some authors. Little work has been presented on the transient analysis of $M(t)/E_k(t)/n$ or $M(t)/E_k(t)/n/n+q$ models, perhaps due to the difficulty in generating solutions for the systems. Our results indicate ELC may be useful in conducting a more thorough analysis of such systems, particularly using the expected number of customers in the system.

4.3 Validation: Dynamic Parameters

The results in Section 4.2 motivated the analysis of the heuristic solution technique under nonstationary conditions. In this section, we present results of several examples using time-dependent arrivals to the system, $\lambda(t)$.

We are interested in analyzing the models during the transient period as well as during the dynamic steady-state, if one exists. The purpose of this analysis is to provide a preliminary validation of heuristic solution technique under nonstationary parameters by comparing behavior generated using ELC with the exact behavior. As in the previous section, the epoch size is a minute, the systems have limited capacity and start empty and idle. We present results for the expected number of customers in the system and the expected virtual delay, as defined in Section 3.2.

Two models are presented in this section. The first is the $M(t)/E_3(t)/18/18+3$ queueing system, with an arrival rate given by

$$\lambda(t) = \sin\left(\frac{2\pi t}{30}\right) + 1.16$$

shown in Figure 4-10. The choice of the arrival rate was such that $\lambda(t) > 0$ for all t, to avoid "negative" arrivals to the system. The maximum utilization ratio in the example is

$$\rho = \frac{2.16}{0.1(18)} = 1.2$$

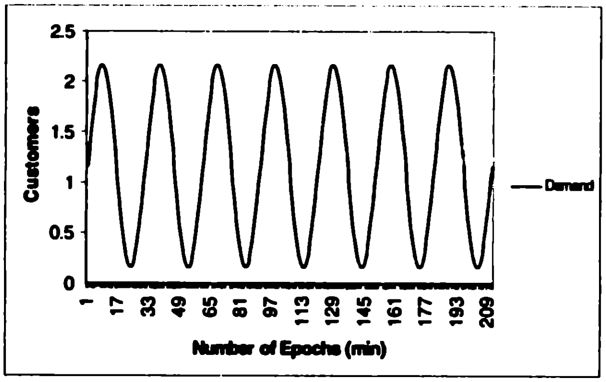


Figure 4-10: Input demand $\lambda(t) = \sin(\frac{2\pi t}{30}) + 1.16$

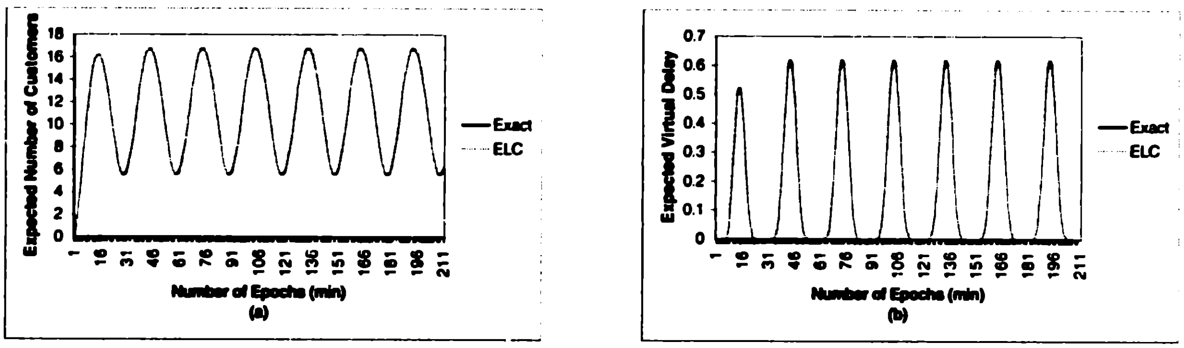


Figure 4-11: $M(t)/E_3(t)/18/18+3$ queueing system with $\lambda(t) = \sin(\frac{2\pi t}{30}) + 1.16$: (a) L and (b) $E[Delay]$ (minutes)

and the minimum

$$\rho = \frac{0.16}{0.1(18)} = 0.088.$$

The results for this system are presented in Figure 4-11. Note that the results using the heuristic solution technique follow almost exactly the results from the exact solution technique in the transient period as well as in the dynamic steady-state. The performance of ELC is as good in approximating the expected number of customers in the system as in approximating the expected virtual delay. A more detailed view of the results in the initial epochs after the system started is presented in Figure 4-12. Even in this zoomed view of the initial epochs of operation, there are no significant differences for practical purposes between the results of both solution techniques. The maximum percentage difference was 2.03% for the expected number of customers in the system, given in epoch 30. In the case of the expected virtual delay, the maximum

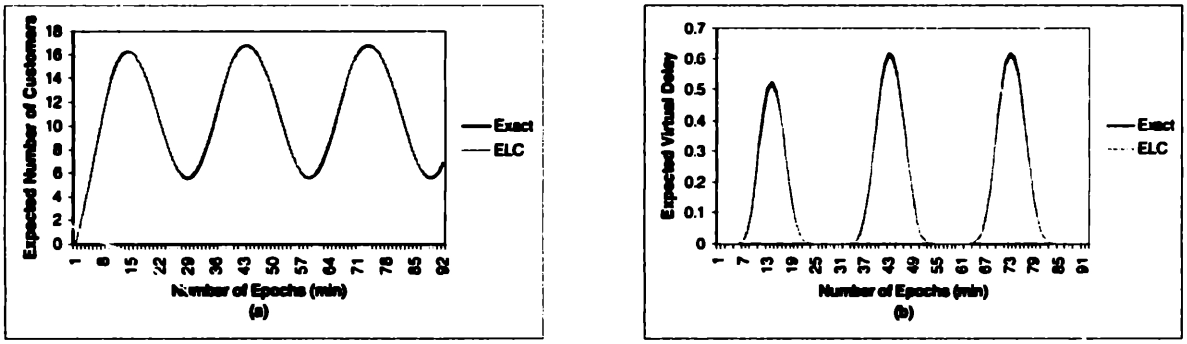


Figure 4-12: $M(t)/E_3(t)/18/18+3$ queueing system (zoom) with $\lambda(t) = \sin(\frac{2\pi t}{30}) + 1.16$: (a) L and (b) $E[Delay]$ (minutes)

percentage difference was 23.81%, in epoch 61, and periodic every 60 epochs, but the actual difference was less than $\frac{1}{1000}$ of a minute. Most percentage differences were within 2% of the exact results in both performance measures.

The second example is the $M(t)/E_5(t)/5/5+15$ queueing system. The arrival rate for this system is given by

$$\lambda(t) = \frac{1}{2}\sin\left(\frac{2\pi t}{40}\right) + 0.75$$

shown in Figure 4-13. The maximum and minimum arrival rates are 1.25 and 0.25 customers per minute, giving utilization ratios of $\rho = 2.5$ and 0.5, respectively.

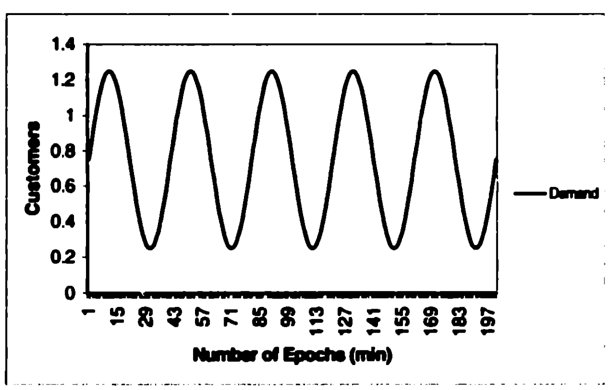


Figure 4-13: Input demand $\lambda(t) = \frac{1}{2}\sin(\frac{2\pi t}{40}) + 0.75$

Results using the exact and heuristic solution techniques for this system are presented in Figure 4-14. Figure 4-14 illustrates behavior from start until the system is in dynamic steady-state. As in the previous example, the results using both solution

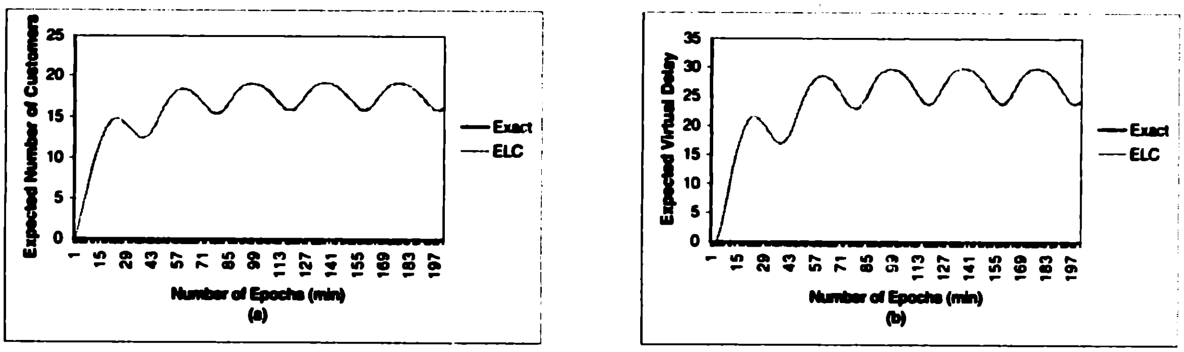


Figure 4-14: $M(t)/E_5(t)/5/5+15$ queueing system with $\lambda(t) = \frac{1}{2}\sin(\frac{2\pi t}{40}) + 0.75$: (a) L and (b) $E[Delay]$ (minutes)

techniques are practically equal, showing that ELC is an excellent approximation of the exact values. In this case, the maximum percentage differences for the expected number of customers in the system and the expected virtual delay are 1.6% and 3.33%, respectively. Most values for both performance measures are within 2% of the exact results.

In this section, we have shown preliminary evidence that the heuristic solution technique provides an excellent approximation of the exact results of the $M(t)/E_k(t)/n/n+q$ queueing systems with nonstationary parameters. The importance of these results is that we can use the heuristic for a wide variety of applications with time-varying arrival and service rates, like the application described in Chapter 5.

With this section, we have completed the validation of the heuristic solution technique against the exact results for the case of steady-state, transient analysis and dynamic behavior. In all three cases, we have provided enough evidence to conclude that ELC is very accurate, much faster than the exact model and very reliable for the type of parameters used. In the next section, we present some intuitive results to show the response of the system to a variable number of servers. Section 4.4 is not part of the validation since no exact results are available.

4.4 Results for Systems with Changes in the Number of Servers

In this section, we explore the behavior of the $M(t)/E_k(t)/n(t)/n(t) + q$ queueing systems using Heuristics 3 and 4 presented in Section 3.3 for both the exact and the heuristic solution techniques. We analyze several models comparing the results for the exact and ELC results, and provide an intuitive explanation for the behavior of the system after the number of servers is changed. We present here the results of five such examples, each with one or two changes in the number of servers.

Notice that during the transient time when a system is modified into a new system, the results provided using Heuristic 3 for the exact solution technique are not really exact. Since we are using a heuristic approach to map from the original system to the modified system, no truly exact results can be generated.

The measures of performance that we discuss in this section are the probability of saturation, the expected number of customers in the system and the expected virtual delay. We are interested in analyzing the congestion of the system when changes in the number of servers occur, and believe that those measures of performance summarize this effect. One aspect in which we are particularly interested in is the probability of rejection of customers that arrive at the system.

As in the previous sections, we solve the Chapman-Kolmogorov equations for finite queue size systems. In this section, we do not solve examples with large enough waiting room to achieve effectively infinite capacity systems. One reason for not doing that is because we are interested in finding the rejection probabilities of the queueing systems under investigation. This interest is mainly motivated by the practical application presented in Chapter 5. The second reason is because systems with high utilization ratios, or slightly over-saturated systems, may become extremely over-saturated when the number of servers is reduced.

Since we want to observe the systems frequently, the size of the time periods in the examples presented is equivalent to one minute per time period, assuming that a server which is busy for 60 time periods, processes an average of 6 customers in

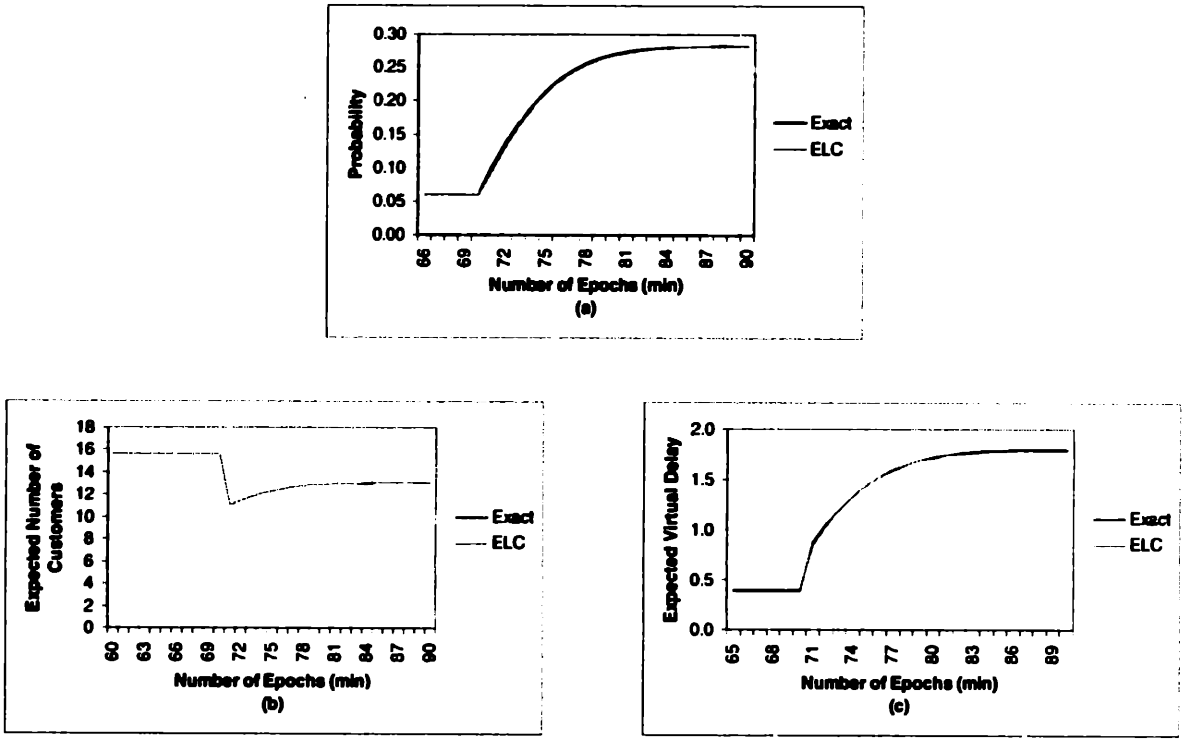


Figure 4-15: Change from $M/E_3/18/18+3$ to $M/E_3/12/12+3$: (a) $P(Saturation)$, (b) L, and (c) $E[Delay]$ (minutes)

that time. We show the transients when the number of servers changes, and observe the system from before any change occurs until it reaches steady-state after the last change in the number of servers. All systems start empty and idle.

We are not interested here in validating the performance of a heuristic solution technique against an exact solution technique. We are interested in validating Heuristics 3 and 4 proposed in Section 3.3 against one another through intuitive analysis of the exact and ELC results. The heuristics to modify the number of servers, increase or decrease the number of servers by only one at a time. If more than one servers are to be closed or opened, the heuristics can be iterated to obtain the desired final number of servers in the system.

The first set of results is for the example starting with an $M/E_3/18/18+3$ queueing system and changing to an $M/E_3/12/12+3$ model. The results for this example are in Figure 4-15. The utilization ratio for the initial system is $\rho_1 = 0.9$, with an arrival rate to the system of $\lambda = 1.62$ customers per time period, and a service rate per server of $\mu = 0.1$ customers per time period. The corresponding utilization ratio for the modified system, with the same arrival and service rates, is $\rho_2 = 1.35$. Note that the probability of saturation increases considerably, as expected, when the number of servers is reduced from 18 to 12 (see Figure 4-15, (a)). In the original system, we are rejecting approximately 0.1 customers per minute, while in the modified system we reject approximately 0.45 customers per minute. Figure 4-15, part (b) shows the results for the expected number of customers in the system. The expected number of customers in the system decreases sharply initially from about 16 customers to about 11, a jump that is due to the number of servers removed. The probability of having more than 15 customers is now zero. After the change in the number of servers, the expected number of customers in the system increases until it reaches steady-state at approximately 13 customers. The short term effect of ignoring the statistics of the customers in the servers that are closed is evident in this sharp decrease in the expected number of customers in the system. If we would account for those statistics, we would expect a smoother change which would eventually reach the same steady-state value of the modified system. In the same Figure, part (c), we present the results for the expected virtual delay. In this graph, we observe that the steady-state expected delay for the modified system is approximately 4 times the steady-state expected delay of the original system. The transient period does not show any sharp change, similarly with the probability of saturation, because the probabilities of having customers in the queue are small in the original system and start increasing once the system has closed 6 servers, as the utilization ratio increases considerably.

We can see that Heuristics 3 and 4 used to modify the number of servers in the system provide reasonable results. It is expected that the rejection probability and the delay for customers entering the system will increase as the number of servers is reduced while maintaining the same arrival and service rates. We also expect that the number of customers in the system will change as there are fewer servers available and the capacity of the queue remains the same. From the point of view of the customers in the queue and those arriving to the system, the heuristics provide realistic statistics as they see the system with only the reduced number of servers. For the customers in

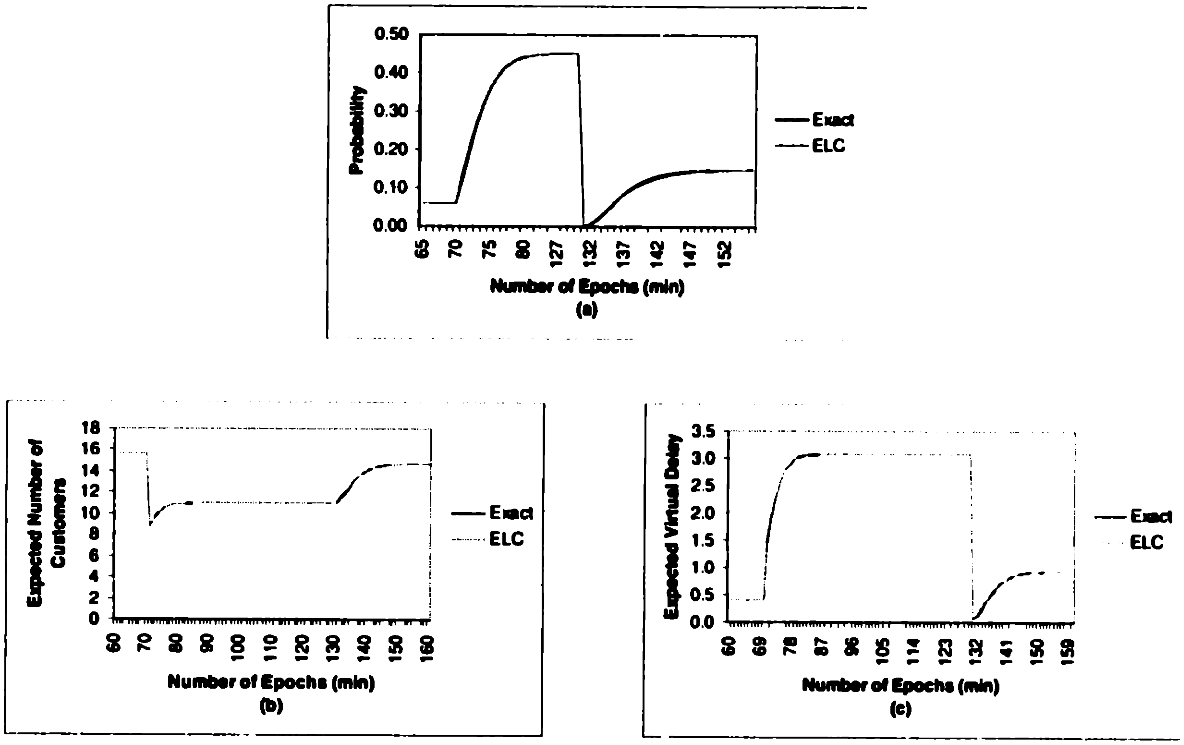


Figure 4-16: Change from $M/E_3/18/18+3$, to $M/E_3/9/9+3$ and to $M/E_3/15/15+3$: (a) $P(Saturation)$, (b) L, and (c) $E[Delay]$ (minutes)

service, they are only waiting for their service to be finished and they are not affected by the change in the number of servers.

The results in the example above are obtained using the exact and heuristic solution techniques. Both solution techniques provide extremely similar results. We observe the same situation in all the examples presented in this section. The fact that both sets of results are so close suggests that initializing the patterns that cannot be mapped, when using Heuristic 4, is not a significant practical or evident limitation of the heuristic. (The problem of initializing patterns that cannot be mapped when changing the number of servers in the system was discussed at the end of Chapter 3.)

We continue with example 2. In this example, we study the transition from the $M/E_3/18/18+3$ model to a system with one half the number of servers, the $M/E_3/9/9+3$ system, and then to the $M/E_3/15/15+3$ model. Figure 4-16 shows the results for this example. The initial utilization ratio is $\rho = 0.9$, with the arrival and service rates $\lambda = 1.62$ and $\mu = 0.1$ customers per minute, respectively. Those rates are kept constant and the utilization ratios for the second and third systems are

1.8 and 1.08, respectively. The highly over-saturated second system rejects almost half of the customers that arrive to the system, once it reaches steady-state. The probability of saturation for this example is presented in Figure 4-16, part (a), and we see that it increases in an exponential-like curve when the system is modified for the first time. For the second change, the probability of saturation reduces to zero since we are adding servers to the system and in the instant the servers are added, the customers in the queue enter service and the probability of having 18 customers in the system is initialized to zero. Then, the probability of saturation increases to reach steady-state. The expected number of customers, Figure 4-16 part (b), shows a similar behavior as in the example 1 when the number of servers is reduced from 18 to 9, and the steady-state value is close to the maximum number of customers that can be in the system at any instant of time. When 6 servers are added again to the system, the expected number of customers increases but without being as saturated as in the case with 9 servers since the system is not as over-utilized. The effects of varying the number of servers are also evident in the expected virtual delay. Note that the expected delay increases more than 6 times when we reduce the number of servers. With the increase in the number of servers in the second modification, we observe that the expected delay goes to zero and then increases to a steady-state value of almost a minute. The reason for the drop to zero is because at the moment we add the servers, the heuristics initialize to zero the probabilities of having 13 to 18 servers in the system causing the expected delay to be zero at that time (the probabilities of 16, 17 and 18 customers in the system are responsible for delays incurred). The number of servers in the final system is larger than the total capacity (in service and in the queue) of the intermediate system.

The third set of results is for a system with nonstationary arrival rates. The initial system is an $M(t)/E_3/18/18+3$ queue that changes to an $M(t)/E_3/9/9+3$ queue. In this example, the arrival rate is given by

$$\lambda(t) = \sin\left(\frac{2\pi t}{30}\right) + 1.16$$

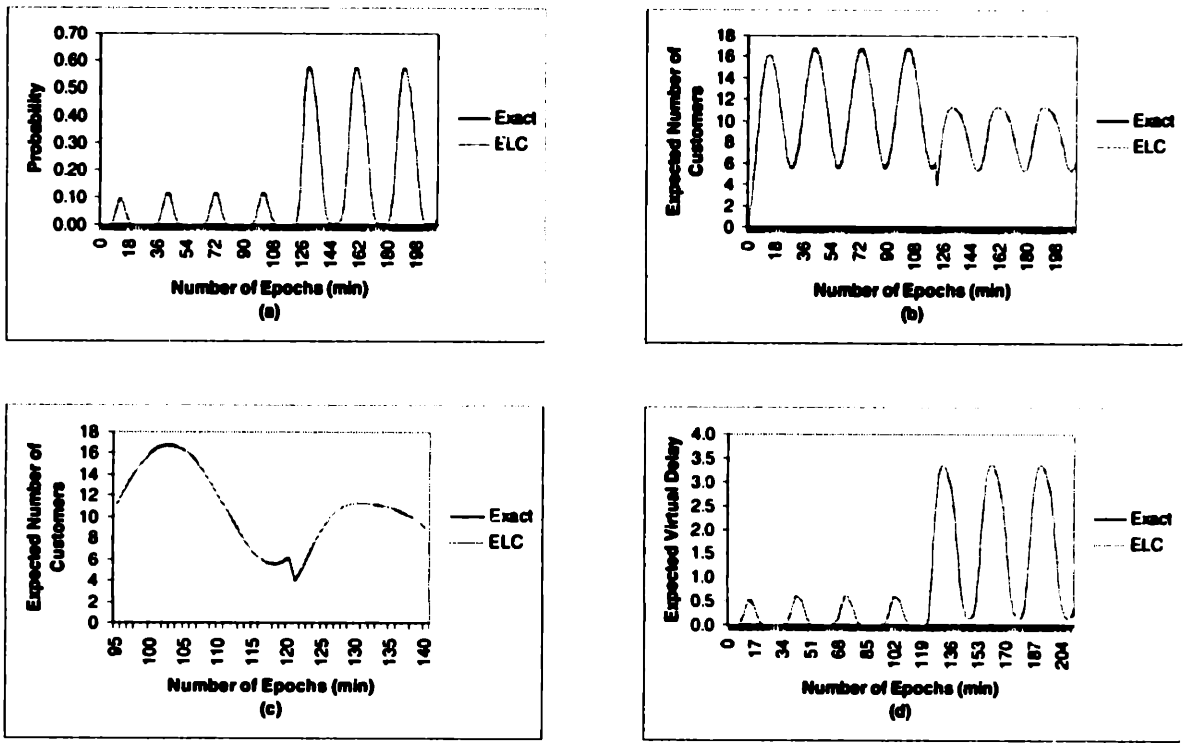


Figure 4-17: Change from $M(t)/E_3/18/18+3$ to $M(t)/E_3/9/9+3$: (a) $P(Saturation)$, (b) L, (c) zoom of L, and (d) $E[Delay]$ (minutes), with nonstationary arrival rate

shown in Figure 4-10, and the service rate is $\mu = 0.1$ customers per minute. The maximum and minimum utilization ratios for the original and modified systems are (1.2,0.088) and (2.4,0.177), respectively. The results for this example are shown in Figure 4-17. We observe the same type of behavior as in the previous two examples when the number of servers is reduced: the probability of saturation increases, the expected number of customers is reduced due to the very limited queue size and the expected virtual delay increases accordingly. Interestingly, as the number of servers is reduced to 9, the expected number of customers in the system is reduced by a small amount, due to the low utilization ratio given by the low demand at the lower part of the sinusoidal input. This can be seen in Figure 4-17, parts (b) and (c). Figure 4-17 is a magnified view around the time the servers are closed. In the previous examples, with stationary parameters, we had larger drops because the utilization ratios were high at the moment the number of servers decreased. The high peaks in the expected number of customers in the system also become flatter as the system reaches its maximum capacity for the number of customers in the system.

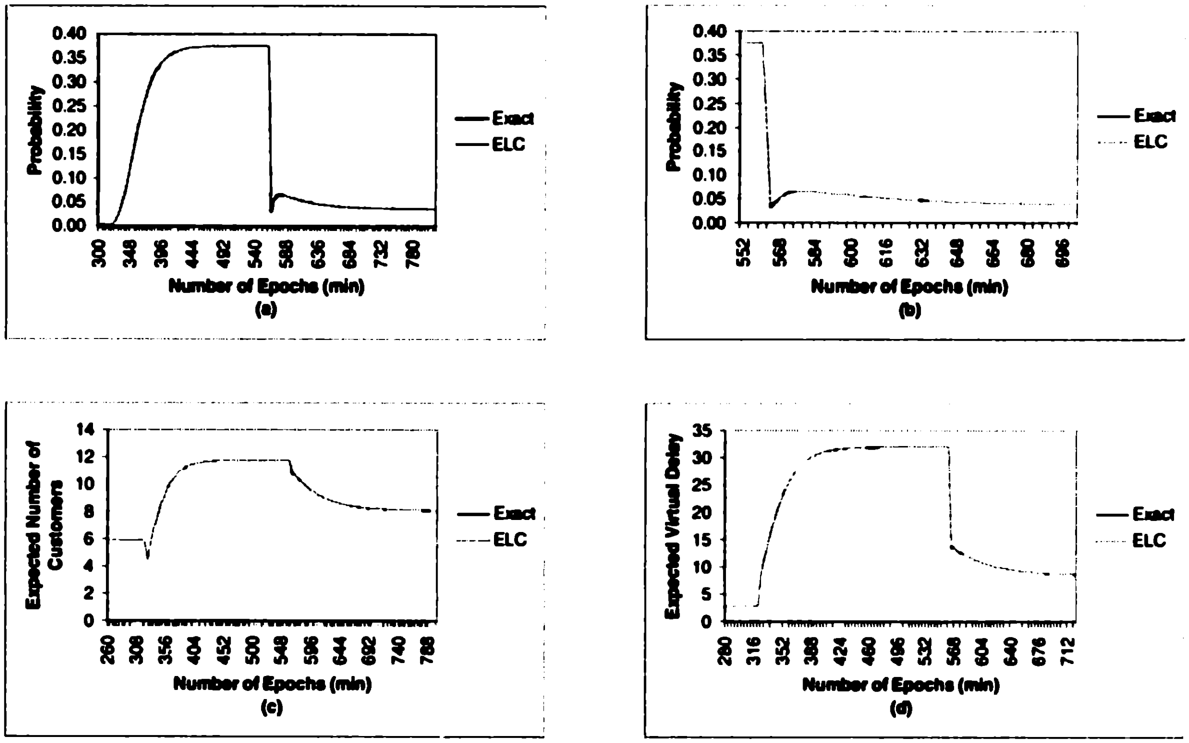


Figure 4-18: Change from $M/E_4/6/6+10$, to $M/E_4/3/3+10$ and to $M/E_4/5/5+10$: (a) $P(Saturation)$, (b) zoom of $P(Saturation)$, (c) L, and (d) $E[Delay]$ (minutes)

We have examined so far in much detail examples for systems with Erlang order $k = 3$, queue size $q = 3$ and several number of servers n. The reason for such an emphasis is that the application presented in Chapter 5 uses those systems to model the en-route sectors in the airspace. The next two examples have different Erlang order and a larger buffer for customers waiting to enter the service facility.

Example 4 starts with the $M/E_4/6/6+10$ system, which is later modified to $n = 3$ and finally to $n = 5$. The arrival and service rates were kept constant at $\lambda = 0.48$ customers per minute, and $\mu = 0.1$ customers per minute, respectively. The initial utilization ratio is $\rho_1 = 0.8$, changing to $\rho_2 = 1.6$ and finally to $\rho_3 = 0.96$. The results for example 4 are shown in Figure 4-18. The probability of saturation has an interesting behavior (see Figure 4-18, (a) and (b)). Because of the larger buffer, the initial rejection probability is almost negligible, and it increases considerably as the number of servers decreases. The second modification in the system causes the probability to drop to zero (even though this is not obvious in Figure 4-18 (b) because we plot the results every 4 minutes) and then over-shoots before reaching steady-state.

The over-shoot may be a consequence of having such a large probability of saturation and large probabilities of having customers in the queue, that as we increase the number of the servers, we have a transient time in which those probabilities remain high causing the new probability of saturation to increase and later to reach its final value. In the case of the expected number of customers in the system, Figure 4-18 part (c), we observe that the system experiences a drop in the number of customers right at the time the servers are closed, but it then starts increasing because of the high utilization ratio and the larger queue size. In the previous examples we observed a reduction in the expected number of servers in the system due to the small waiting room. After the second modification, the expected number of customers is reduced mainly because the utilization ratio is considerably smaller. The expected virtual delay increases after the first modification of the system, as in the examples before, but it does not drop to zero after the second modification. The reason for not dropping to zero is because some customers that were in the queue in the intermediate system remain in the queue even after the number of servers increases.

The last example of this section, example 5, presents the results for the transition from the $M/E_{10}/3/3+10$ system to the $M/E_{10}/2/2+10$ system, and finally to the $M/E_{10}/4/4+10$ system. The behavior of example 5 is similar to the behavior of example 4 since both have a larger queue size and a small number of servers. The utilization ratios for this example are $\rho_1 = 0.8$, $\rho_2 = 1.2$ and $\rho_3 = 0.6$, with a constant arrival rate $\lambda = 0.24$ customers per minute and service rate $\mu = 0.1$ customers per minute. The results of this example are presented in Figure 4-19. The probability of saturation for the initial and final systems are very small because of the low utilization ratio. In Figure 4-19 part (a) we observe that the rejection probability is reduced sharply, almost to zero, as the number of servers increases and it eventually becomes zero in steady-state. Similarly as in example 4, the expected number of customers in the system and the expected virtual delay increase in the intermediate system and decrease after the second modification of the system.

As an additional check, for all examples presented in this section, we have compared the steady-state values for all initial, intermediate and final systems with the

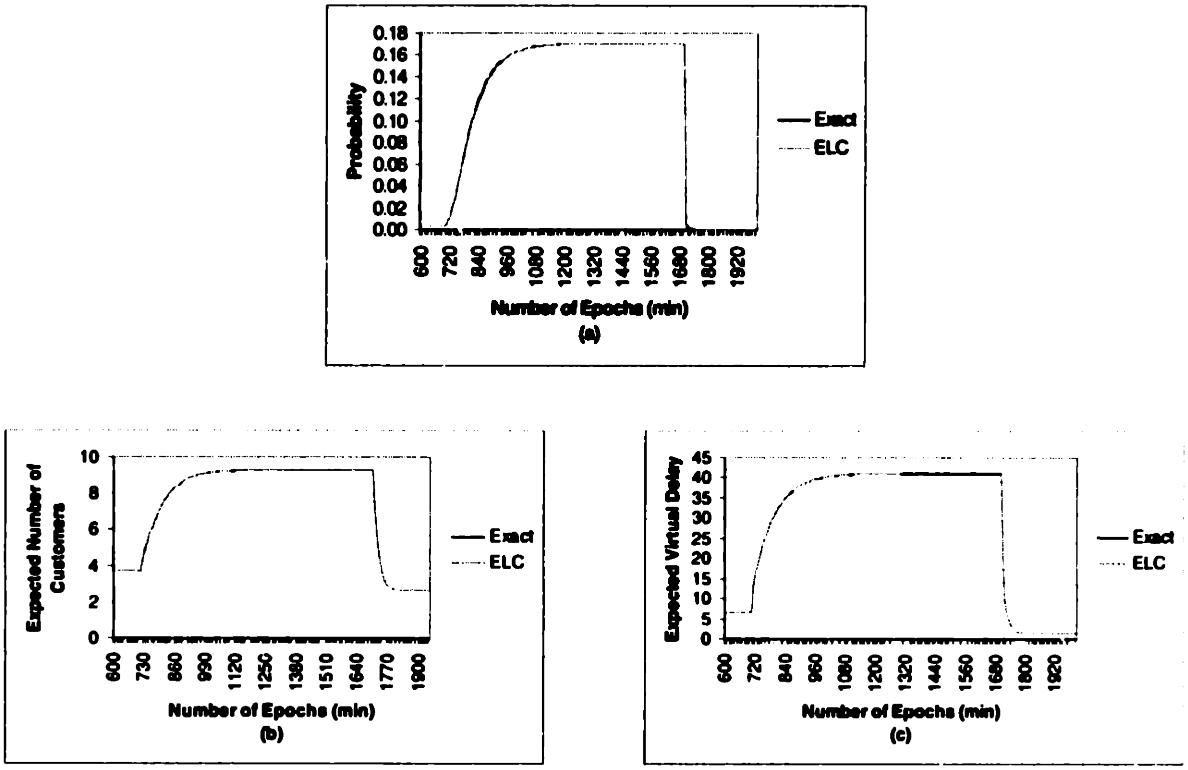


Figure 4-19: Change from $M/E_{10}/3/3{+}10$, to $M/E_{10}/2/2{+}10$ and to $M/E_{10}/4/4{+}10$: (a) $P(Saturation)$, (b) L, and (c) $E[Delay]$ (minutes)

steady-state values of corresponding systems that have been running, without changes in the number of servers, and with the same parameters. All the results were identical, confirming our intuition that after the transients have died down, the modified systems behave as if they had been running without changes in n all the time. We can also conclude that if the statistics of the customers in the servers that are removed are not necessary for the analysis of the system, Heuristics 3 and 4 proposed in Section 3.3 adequately capture the behavior of the system with a variable number of servers. We have observed that the results for systems with variable n depend greatly on the utilization ratio and the size of the queue.

4.5 Conclusion

We have shown through an extensive set of examples and scenarios that the heuristic solution technique, ELC, is an excellent approximation of the exact results, with a wide range of system parameters. Therefore, a more thorough analysis of such systems can be carried out with ELC instead of solving the exact system. An interesting future research topic with $M(t)/E_k(t)/n(t)/n(t)+q$ systems would be to determine the transient times required to reach steady-state.

At this point, we are ready to present a practical application of the $M(t)/E_k(t)/n(t)/n(t)+q$ queueing systems. In Chapter 5, we describe the implementation of the exact and ELC solution techniques, and Heuristics 3 and 4 to modify the number of servers in the system. We also present a case study for the behavior of en-route sectors in the airspace, under different scenarios, using our model.

Chapter 5

Computer Models and an Application to Air Traffic Management

This Chapter begins with a description of the computer programs developed to implement the exact and heuristic solution techniques discussed in Chapter 3. Then, we present a case study to illustrate the use of these techniques in the context of Air Traffic Management. More specifically, we address the modeling of high altitude sectors of the U.S. airspace with $M(t)/E_k(t)/n(t)/n(t)+q$ queueing systems. The case study explores multiple scenarios with future demand forecasts and capacity fluctuations, common in the presence of changing weather, and includes a baseline case with data for a particular sector on April 8, 1996. Using those scenarios, we analyze their impact on expected delays and rejection rates for users of the sector. We also study the workload of the air traffic controllers handling the sector.

5.1 Computer Hardware and Software

The software developed was run on a SUN SPARCstation 10 Model 41. We used the Inter-Math-Science-Libraries (IMSL) ordinary differential equations (ODE) solver to solve the Chapman-Kolmogorov equations of the $M(t)/E_k(t)/n(t)/n(t)+q$ queueing systems. IMSL uses a fifth and sixth order Runge-Kutta method in solving the ODE's, with a global error tolerance of 10^{-6} per call to the ODE solver.

All computer programs were written in the C and C++ programming languages. The algorithms were developed by the author of the thesis and some were partially implemented by him. Most of the computer programs were written and optimized by Mr. Wesley McDermott, MIT SM'94.

5.1.1 Implementing the exact solution technique: *exact-model*

We now describe the software developed to solve the Chapman-Kolmogorov equations for the exact solution technique. The required inputs to the program, the structure of the program itself and the outputs generated are presented below.

The system parameters needed as input are the Erlang order k, the queue capacity q and a vector with the number of servers, $\mathbf{n}$, at each epoch. An epoch is the unit of time at which we observe the system. One restriction to the Erlang order is that $k \geq 2$. The case for $k = 1$ is solved with the *mekn-model* described below (Section 5.1.2) as both programs generate the same ordinary differential equations. Figure 5-1 shows the flow diagram for the *exact-model*. In the input data, we also must specify the values or type of functions of the arrival and service rates, i.e. if they are constant or time-varying, and if we would like to interpolate between the different values of $\lambda(t)$ and/or $\mu(t)$ for contiguous epochs in the case the rates are given for each epoch instead of by a function of t. The last two input quantities needed to run the program are the unit of time corresponding to each epoch and the number of epochs of the experiment. Once the input parameters have been indicated in the code, the program is compiled using a C++ compiler. Every time we modify any parameter, the program needs to be re-compiled.

The program generates the state probability arrays $\mathbf{S_0}$ and $\mathbf{Q_i}$'s, $0 \leq i \leq q$, and maps the elements in the arrays that can be reached with the transitions in Table 3.2, into a vector with all the state probabilities. Once the vector is generated, the state

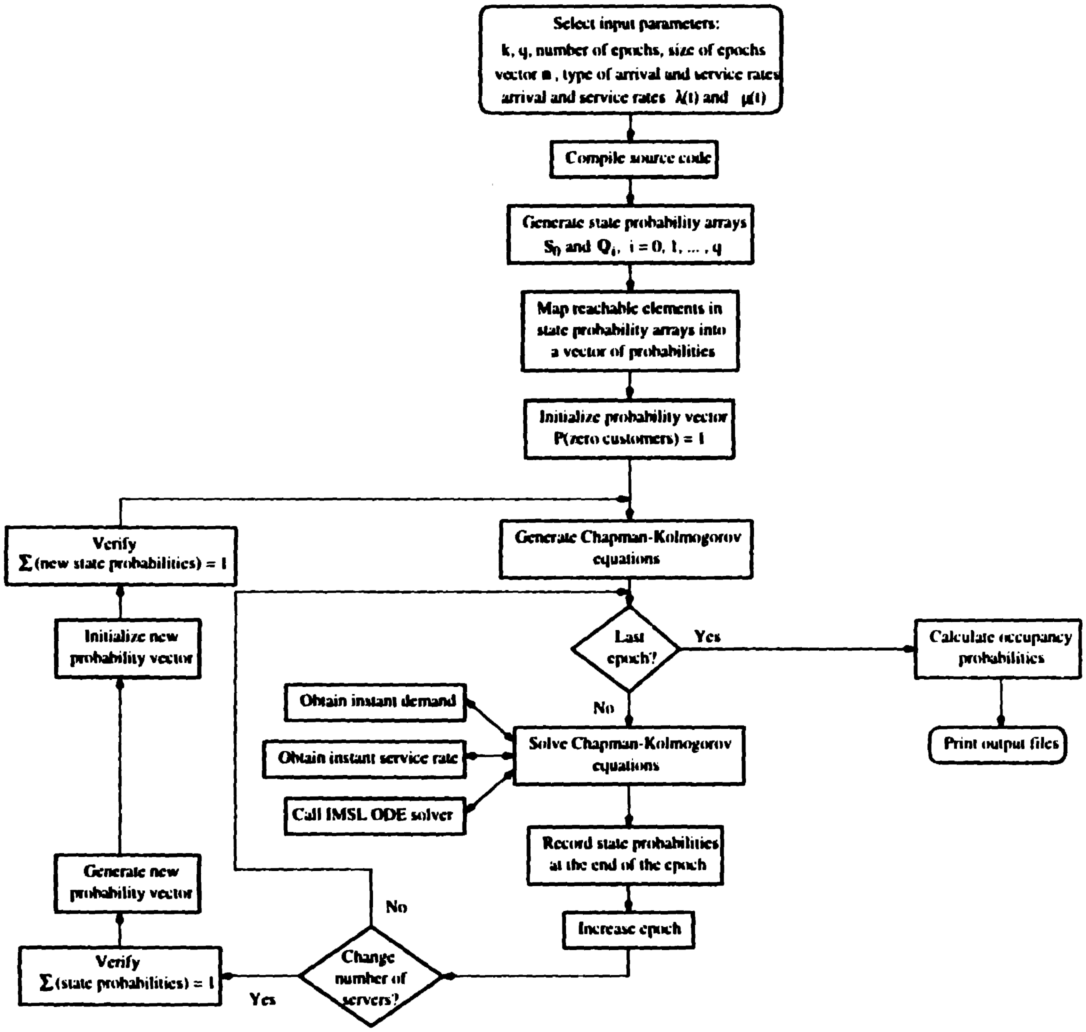


Figure 5-1: Flow diagram for *exact-model*

probabilities are initialized to zero, except the probability of having zero customers in the system which is initialized to one. Then, the Chapman-Kolmogorov equations in Section 3.1.3 are generated for the parameters given and are solved using the ODE solver of the IMSL software. The main program calls the ODE solver and the subroutines for the arrival and service rates every time the equations are solved. The subroutines for the arrival and service rates compute the values for $\lambda(t)$ and $\mu(t)$, respectively, for all t required by the Chapman-Kolmogorov equations. If the number of servers changes from one epoch to the next, the main program verifies that all state probabilities are mapped correctly and that the sum of all state probabilities is equal to one, before and after the change in the number of servers. It also initializes to zero the state probabilities that are not mapped when the number of servers increases.

When the program finishes, it outputs the state probability distribution at the end of each epoch. With the state probabilities, we compute the occupancy probabilities of Section 3.2 needed to obtain the desired performance measures described in Sections 3.2.1 and 3.2.2. The program also provides the CPU time used in computing the state probability distribution and the distribution of customers in the system.

5.1.2 Implementing the heuristic solution techniques: *mekn-model*

To solve Chapman-Kolmogorov equations for the heuristic solution technique, we developed a program similar to *exact-model* in structure and requirements. This program needs an additional input and an additional process to generate all the parameters in the ordinary differential equations.

The *mekn-model* requires as input the system parameters k, q and $\mathbf{n}$, the number of epochs, the size of an epoch, and the time-varying arrival and service rates $\lambda(t)$ and $\mu(t)$, described in the previous section, as well as the selection of the heuristic solution technique to use: ELP or ELC. The choice of ELP or ELC determines the appropriate transition probabilities α's and β's to use in the Chapman-Kolmogorov equations (Section 3.1.4). Computing these transition probabilities is the extra process required by the *mekn-model.* All these quantities should be defined in the source file which is compiled with a C compiler. As with the *exact-model*, we need to re-compile the program after any change in the input parameter. Figure 5-2 illustrates the flow diagram for the *mekn-model.*

The program generates the state probability array $\mathbf{P}$ and the elements that can be reached with the transitions described in Table 3.3 are mapped to a vector with all state probabilities. The state probabilities are initialized to have zero customers in the system, with probability one. The values for the transition probabilities α's and β's are computed directly in the *mekn-model* and used in generating the Chapman-Kolmogorov equations in Section 3.1.4. The remaining part of the program follows

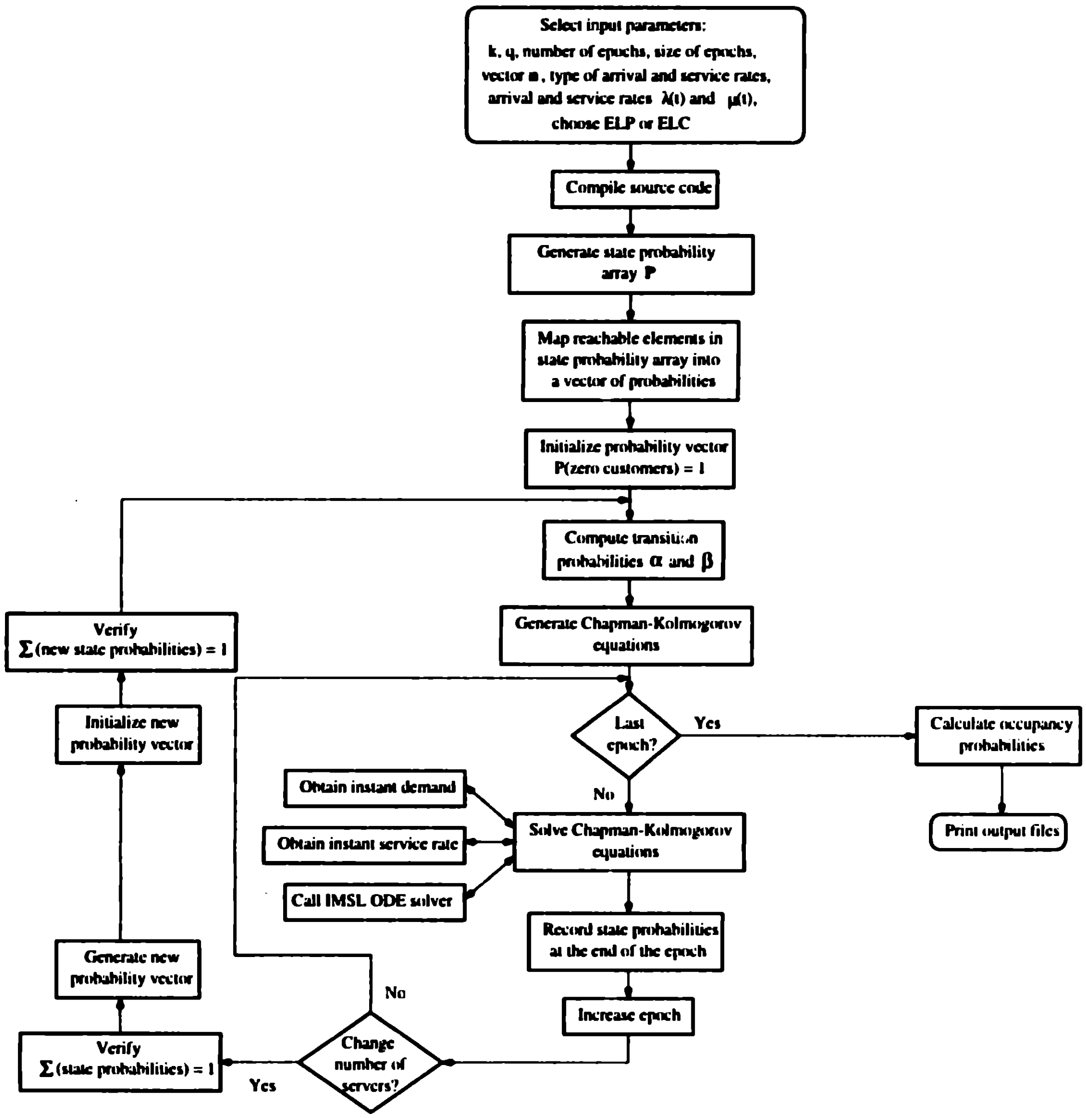


Figure 5-2: Flow diagram for *mekn-model*

the same structure as the *exact-model*: it uses the ODE solver in IMSL with the appropriate arrival and service rates and the number of servers for the epoch being solved. The program uses the same procedure as the *exact-model* to verify that the change in the number of servers is correct.

The *mekn-model* generates the corresponding state probability distribution which is used to calculate the distribution of customers in the system. It also obtains the same measures of performance as those evaluated in the *exact-model.*

5.2 Application to Air Traffic Management

In recent years, many studies have examined the problem of air traffic congestion, which has become endemic in the United States, Western Europe and the Pacific Rim (see [27] for many references). As a result, a number of approaches to modeling that congestion have been presented and several alternative methods for reducing delays and the attendant delay costs and safety costs have been examined in considerable depth (e.g., Ground Holding Policies: [34], [38], [39]).

Most of this work has focused on airport-related congestion which, at least in the United States, currently accounts for the great majority of flight delays. Another source of delays, however, is en-route airspace. Far less effort has been dedicated to date to understanding and modeling congestion in the en-route sectors (e.g., [2]). A model of delays in these en-route sectors would offer the possibility of developing an integrated tool for estimating delays throughout the entire air traffic management system. Such an integrated tool can be developed by combining the new model of en-route sector delays with existing models for estimating delays in a national or regional network of major airports and terminal areas. This model could also help in the problem of re-routing airplanes when one or more sectors are highly congested or closed due to bad weather problems.

The objective of the case study described in this Section is to propose the use of the $M(t)/E_k(t)/n/n+q$ queueing model, with variable number of servers, as a reasonably good model to estimate delays and congestion in an en-route sector.

5.2.1 Basic Operations of an En-route Sector

The description of the basic operations of en-route sectors presented in this Section was compiled through interviews with FAA controllers and managers at the FAA's Air Route Traffic Control Center (ARTCC) in Denver, CO, in conjunction with Dr. David Lee of the Logistics Management Institute (LMI) (see Lee et al. [22]), and from information available in the MIT Lincoln Laboratory report on the operations of the Air Route Traffic Control Center (ARTCC) in Kansas City (see Wilhelmsen at al. [47]).

En-route sectors are classified according to altitude levels into four types: super-high altitude sectors, starting at 33,000 ft. and above; high altitude sectors, which include flight levels between 24,000 ft. and 33,000 ft.; low altitude levels, between 10,000 ft. and 23,000 ft.; and super-low altitude sectors, usually below 10,000 ft.. However, this classification may vary from center to center as can the type of traffic that uses the sector. Low and super-low altitude sectors surround the airspace where most aircraft either maneuver in preparation for an landing or climb after take-off to a higher altitude. Such airspace is called terminal area airspace (TMA). High and super-high altitude sectors are determined primarily by the jet routes and en-route traffic. For example, the ARTCC in Kansas City has 41 sectors: 7 are super-high, 15 are high, 17 are low and 2 are super-low; traffic in the center consists of 50% commercial (of which 70% are over-flights), 25% general aviation and 25% other traffic, including military operations.

The en-route sectors in the air traffic control centers are grouped into geographical areas or regions. In general, center areas are delimited by the sector boundaries of those sectors located on the perimeter of the area. Sector boundaries are drawn to minimize controller workload associated with traffic crossing the boundaries (e.g., avoiding large amounts of traffic crossing only small sections in a sector). At the same time, sector boundaries are designed to balance the workload between sectors, and they are adjusted dynamically and may change during the day. The changes are due mainly to traffic and sector conditions (e.g., bad weather, turbulence, etc.). Air traffic controllers are specialized in one of the center areas and do not work on any other area of the ARTCC. The reason for such controller specialization is the complexity of ARTCC operations.

Our case study presents an example of a high altitude sector. Consequently, we discuss in some detail the basic operations of this type of sector. Traffic arriving to high altitude sectors is generally orderly and predictable, but with periodic daily surges that cause high workload. Aircraft arrival times to the sector are random and the time an airplane spends inside a sector also tends to be random but concentrated around certain values. Airplanes are handled by air traffic controllers without delay until the number of airplanes in the sector reaches maximum capacity. This capacity varies from sector to sector, depending on the traffic characteristics (e.g., climb/descend vs. cruise, intersecting vs. parallel routes, etc.), the sector limitations (e.g., special use airspace, available communications and surveillance, etc.) and the weather. Once the sector is near or at saturation, aircraft are delayed in adjacent sectors by requesting changes in speed and vectoring. There exists a limit on the number of airplanes that can be delayed to enter the congested sector. If the requests exceed the limit, the excess airplanes are diverted to adjacent and less congested sectors.

Air traffic controllers face several challenges when handling high altitude sectors. For example, if a sector has established streams of traffic, controllers need to merge arriving airplanes into such streams; if aircraft passing through a sector are on the way to highly congested airports, controllers have to sequence such airplanes as they approach the terminal area airspace; if there is poor weather in a sector or if a sector is saturated, airplanes have to be re-routed through adjacent and less congested sectors; and, if en-route sectors experience high volume of traffic at high speed, combining such traffic represents a heavy burden on the controllers.

Weather plays a very important role in en-route sector operations. In poor weather, controller workload may be severely affected as more communications with pilots are needed, flight-plans are modified, airplane monitoring increases and interaction with adjacent sectors is higher, among other effects. Even though certain elements of weather are predictable, not all can be determined precisely, e.g., severity and exact location of turbulence, formation of new convective cells and strength of winds aloft.

Another interesting aspect of an en-route sector's operation is the combination of two or more sectors to accommodate staffing needs and the ebb and flow of traffic. For example, in low-traffic hours (late at night), all sectors in a center area may be combined into two sectors: high and low altitude. (In the Kansas City Center, up to 5 sectors may be combined into one.) In high-traffic hours, a controller with a single sector may need the assistance of one or two additional ones in order to handle the high volume of demand.

5.2.2 Modeling an En-route Sector

Let us now summarize the characteristics of en-route sector operations through the following four points:

- Interarrival times to the sector will be assumed for modeling purposes to be independent and exponentially distributed.
- Demand is time-varying.
- Times-in-sector are random, may be concentrated near certain values and are assumed to be independent of each other.
- Sector capacity is variable and there is a limit to the number of airplanes that can wait to enter the sector.

Therefore, a reasonable queueing model to approximate the operations of en-route sectors is the $M(t)/G/n(t)/n(t) + q$ system. The assumptions regarding the interarrival time distribution and the independence of the service time distribution need to be considered carefully.

Determining the true demand of en-route sectors is not an easy task since observing and interpreting the arrival process is difficult and complicated. The arrival data to the sector may be biased by controllers' actions: we cannot fully determine the demand just from the arrival data available because we do not know how many airplanes were diverted or delayed prior to arrival in the sector to enforce en-route sector capacities or to avoid bad weather in the sector, i.e., we can only observe the actual number of planes that crossed the sector in a period of time; and, we may not observe the true demand of the sector since airplanes could be spaced prior to entering the sector to satisfy separation requirements, i.e., aircraft will not arrive in clusters or as frequently if controllers have already spaced them before arrival to the sector. Another approach to determine the en-route sector demand could be by counting aircraft that intend to go through the sector. One problem with this approach is that airplane routes may be biased because airlines may have designed them to avoid highly congested sectors. Therefore, using the assumption that the arrival process is Poisson may be a reasonable approximation but may not be absolutely correct.

At the same time, times-in-sector may be greatly influenced by the particular airways in the en-route sector, since most airplanes stay on the airways throughout the sector, fly at speeds within a certain range and need to maintain separation requirements with other airplanes on the same airway. Thus, the independent times-in-sector assumption may also be violated. If an en-route sector does not have major airways where almost all airplanes fly, the independent service time distribution may be safely applied since aircraft will follow numerous routes within the sector at various speeds.

In the next section, we present data for the high altitude sector ZID095A, in the Indianapolis Center, which we use to validate our proposed model.

5.3 High Altitude Sector ZID095A

The ZID095A high altitude sector covers part of Ohio, Kentucky and West Virginia. It is in a location where routes connecting several pairs of major airports intersect. The information about arrivals to the sector was obtained from the Enhanced Traffic Management System (ETMS) and was processed at LMI. The arrival data selected correspond to April 8, 1996, a day with good weather and low winds, in which normal operations can be assumed.

In Figure 5-3, we show the arrival information for the 24 hour period. Note that after the number of arrivals increased above 50 aircraft in an hour, it remained high until the end of the day. The hours with the most operations observed are from 9 until 10 am, and from 6 until 7 pm, with maxima of 82 and 87 aircraft, respectively. The total number of aircraft observed passing through the sector during the day were 1,149. As we said before, the number of aircraft observed in the sector may not correspond exactly to the true demand as air traffic controllers may have diverted some aircraft to other sectors.

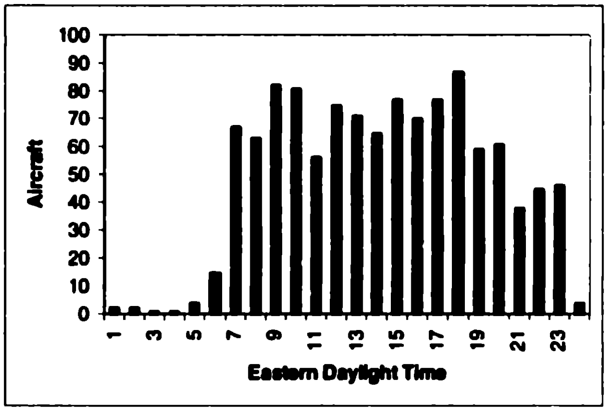


Figure 5-3: Arrivals to ZID095A on April 8, 1996

The Poisson arrival assumption was checked by LMI by comparing histograms of interarrival times with those of a Poisson process. A sample of those comparisons is shown in Figure 5-4 with the interarrival times for aircraft crossing the ZID095A sector between 10 and 11 am. There were a total number of 81 arrivals during that period of time. The arrival rate of 83 aircraft per hour for the Poisson process minimizes the sum of the squares of the differences between the data and Poisson histograms. The classical Chi-squared test for goodness of fit shows that we would accept (fail to reject) the hypothesis that the arrival information has the Poisson distribution with confidence greater than 98 percent. Figure 5-4 shows the number of arrivals between intervals of 0.6 minutes, starting from zero, for both the actual interarrival times during the sample period and the Poisson arrival process.

The service time distribution for aircraft in the sector is obtained from the times-in-sector of the planes that crossed the sector. Figure 5-5 shows the number of planes that spent between 0 and 5 minutes in the sector, 5 and 10 minutes, 10 and 15 minutes, and so on. The mean and standard deviation are 13.8 and 5.9 minutes, respectively. The data of Figure 5-5 suggest that the distribution of service times can be approximated with an Erlang distribution. According to ARTCC controllers, the sectors are designed to eliminate "corner-clipping" or flights that stay in the sector for very short times. They also mentioned that a few airplanes stay in the sector for extended periods of time, e.g., military training flights. Such specific features

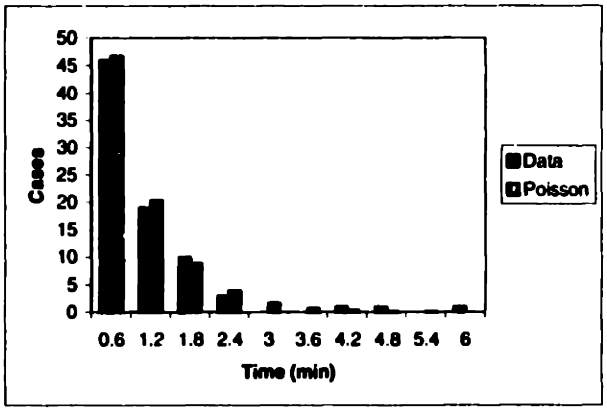


Figure 5-4: Distribution of Interarrival Times for 10 to 11 am and a Poisson process with $\lambda = 83$

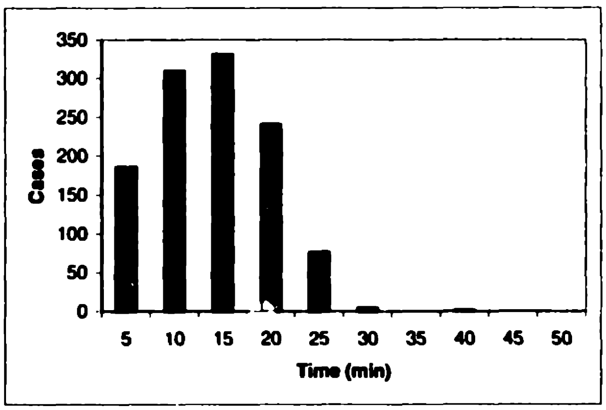


Figure 5-5: Times-in-Sector on April 8, 1996

of transit times are considered in the Erlang distributions' general characteristics. Figure 5-6 shows different curves for Erlang distributions with a mean of 13.8.

In Figure 5-7, we compare the distribution of the transit times in the sector with those of Erlang distributions of orders 3 and 4. Clearly the transit times may be approximated reasonably well with either distribution. Choosing the Erlang order $k = 3$, we minimize the square of the differences between the actual times-in-sector and the Erlang histograms.

Determining the capacity of the en-route sector from the ETMS data is not straightforward. One problem is that we cannot observe directly the capacity of the sector because we do not know if the volume of traffic observed was constrained

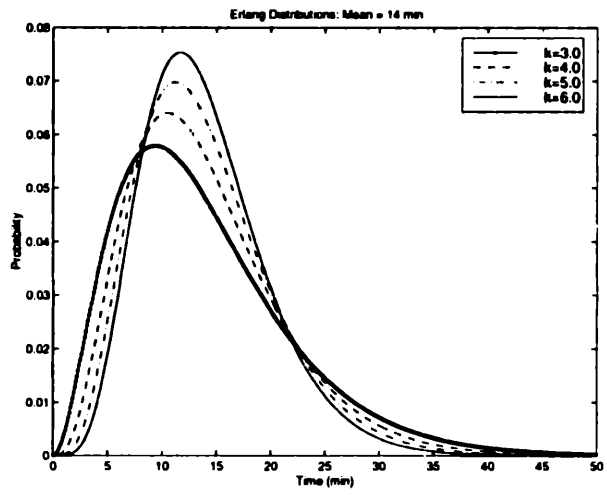


Figure 5-6: Various Erlang distributions with Mean = 13.8

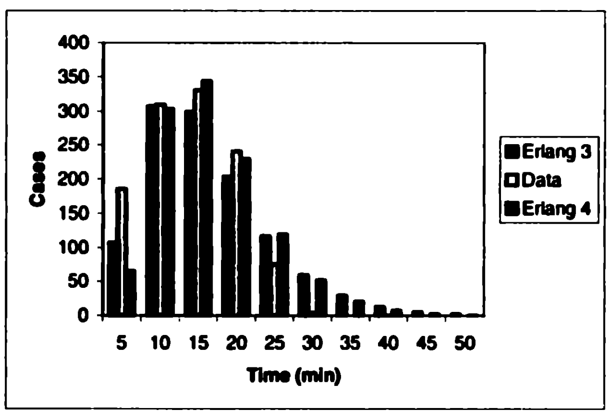


Figure 5-7: Distribution of Times-in-Sector vs. Erlang Distributions of Orders 3 and 4

by capacity limits and because the information in the ETMS data is recorded every 5 minutes. The actual number of airplanes in the sector may be used to determine lower limits on the capacity since we know that controllers were able to handle at least the volume of traffic recorded in the ETMS data. In order to reduce the effect of the 5-minute interval between ETMS reports, LMI used an 11-minute moving average, instead of direct counts from such reports. The result from this analysis was that for the ZID095A en-route sector the capacity lies between 18 and 20 airplanes (see Lee et al. [22] for a detailed description of the analysis). This result is consistent with the values 18 ± 3 implied by reference [1], and with the comments of controllers and managers at the Denver ARTCC. We use the capacity of 18 aircraft in our baseline case.

The information to determine the queue capacity in the en-route sector was also provided by the ARTCC controllers. They suggested a limit of 3 aircraft waiting to enter the sector. These aircraft experience vectoring and/or speed changes in order to accommodate the volume of traffic already present in the sector. If more than three aircraft request passing through a saturated sector, only three are kept in the queue and the others are diverted to adjacent sectors.

It is important to mention that we have analyzed the ZID095A en-route sector with the arrival and service time data for only one particular day. In order to fully justify the use of the exponential distribution for airplane interarrival intervals and the Erlang distribution for aircraft times-in-sector, we would need to obtain and analyze sector data for several other days. Similarly, in order to determine the actual capacity of the sector we need to process ETMS data for more than just one day.

In the remainder of this Chapter, we present an analysis of some hypothetical questions about the ZID095A high altitude sector. First, we provide the baseline case with the original data provided by LMI. Second, we present a scenario with an increase in demand during the afternoon peak hours. The third example illustrates the effects on the sector statistics when the sector capacity decreases considerably during several afternoon hours. We also show a sensitivity analysis of the rejection probability to changes in the size of the waiting queue. The final example illustrates the use of our model to determine the effect of combining various en-route sectors during low-traffic hours, and shows the sensitivity of the rejection probability when the sector capacity is reduced. Except for the baseline case, we do not have real data to compare our results with. Therefore, the purpose of presenting the scenarios described above is to illustrate the flexibility of our model as a powerful planning tool that includes numerous parameter variations. All the examples presented were solved using the ELC heuristic solution technique.

5.3.1 Baseline Results for Sector ZID095A

In our analysis, we consider the sector in isolation without any interaction with adjacent sectors. We evaluate the sector performance using the following statistics:

- Controller expected workload, given by the expected number of aircraft in the sector.
- Probability of diverted aircraft, given by the saturation probability. Airplanes are rejected when the sector is at maximum capacity and the queue is saturated.
- Expected delay for aircraft that are allowed to pass through the sector.

The modeling of the ZID095A high altitude sector was made with an $M(t)/E_3(t)/18/18+3$ queueing system using the interpolated values of the hourly demand shown in Figure 5-3 and a mean transit time in the sector of 14 minutes, which was maintained constant throughout the day. We used the interpolated values of Figure 5-3 to obtain a smooth transition between arrival rates. The mean transit time gives a service rate of $\mu = \frac{60}{14} = 4.2857$ aircraft per hour per server. The maximum hourly capacity is given by $18 \times \mu = 77$ airplanes. Using the ELC solution technique, the number of states in the system is 472 (compared to 1900 if we were using the exact solution technique) and the CPU time required to obtain the sector statistics for the 24 hour period was 71 seconds.

In Figure 5-8, we compare the actual demand and the expected number of aircraft in the sector during the day. Note that the expected workload in the sector stays at an almost constant level during most of the day, specifically, from approximately 7 am until 8 pm. The constant workload during the day suggests that the demand during the day was influenced by the controllers to keep the capacity under 18 airplanes. Figure 5-8 also illustrates that the expected number of aircraft in the sector follows a similar pattern to that of the demand distribution: as the demand increases (decreases), the expected number of airplanes in the sector increases (decreases) as well. The results in all plots are reported at the end of each hour.

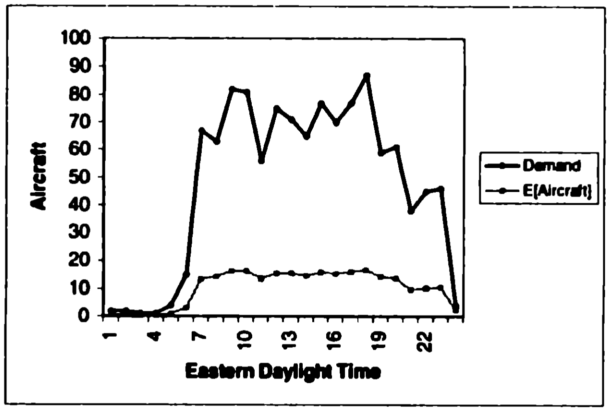


Figure 5-8: Demand and Expected Number of Airplanes in the Sector for April 8, 1996

Even though the expected number of airplanes in the sector is close to saturation during most of the day, the probability that an airplane finds a saturated queue is always less than 0.18 and most of the day is below 0.10. Figure 5-9 shows the probabilities of finding a saturated queue during the entire day. From the 1,149 airplanes that passed through ZID095A, the expected number of aircraft that find a saturated queue is 76.63, with the maximum of 14.91 aircraft between 5 and 6 pm. If we were using scheduled or forecasted demand, those aircraft that find a saturated

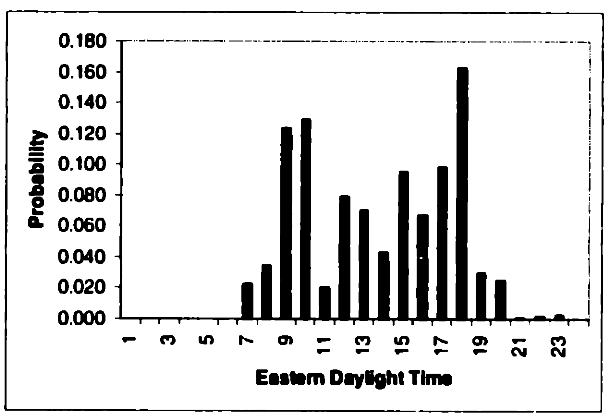


Figure 5-9: Rejection Probabilities on April 8, 1996

queue would have been diverted to adjacent and less congested sectors. The reason for diverting aircraft from congested sectors is to maintain controllers workload within manageable levels.

During normal operations, as those modeled in this baseline case, the expected delay for aircraft that pass through the sector is always less than one minute. The total expected delay during the day for all 1,149 planes is 430 minutes. Figure 5-10 shows the expected delays for airplanes which actually joined the queue and eventually entered the sector on April 8, 1996. Note that the delays in Figure 5-10 do not account for diverted aircraft which may had suffered much longer delays due to diversion. Those amounts of delay incurred by aircraft in the ZID095A sector are reasonable

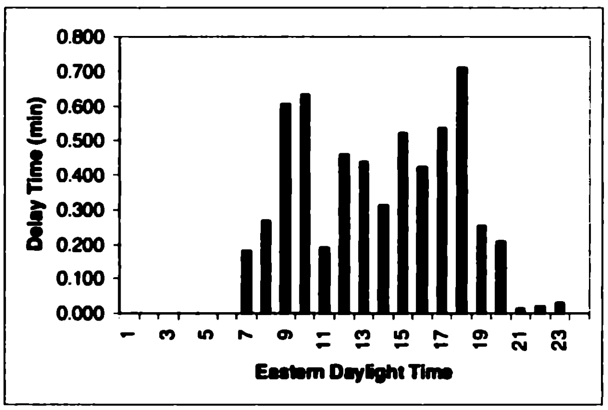


Figure 5-10: Expected Delays for Aircraft Crossing the Sector on April 8, 1996

since most delays in the U.S. air traffic system, during a normal day of operations, are due to congestion at airports. If we were analyzing European en-route airspace, we would expect larger delays because they have tighter constraints in the use of airspace than that at airports.

We presented the above model and results to managers and air traffic controllers at the Denver ARTCC. They agreed that such analysis seems to be a reasonable representation of the behavior of an en-route sector.

5.3.2 Scenario 1: Increase in Demand of Afternoon Flights

We next used an $M(t)/E_3/18/18+3$ queueing model with a service rate of $\mu = 4.2857$ aircraft per hour per server, as before, and with an increased demand of 15% from 12 noon until 7 pm from that shown in Figure 5-3. We also maintained the same sector capacity of 77 aircraft per hour. The system size is 472 states, as in the baseline case. The CPU time needed to solve this example was 73 seconds.

The increase in demand and the resulting change in the controllers' expected workload is illustrated in Figure 5-11. Note that the increase in the demand does not have a significant effect on the controller's expected workload. Instead, the major

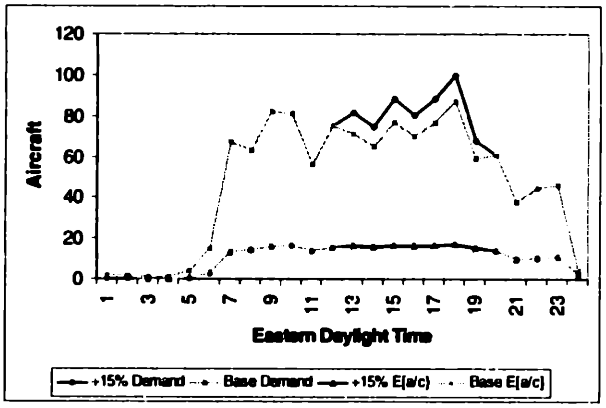


Figure 5-11: Scenario 1: Demand and Expected Workload vs. Baseline

consequence of this increase in demand is reflected in the rejection probabilities and the expected delays, as seen in Figures 5-12 and 5-13, respectively. In Figures 5-12 and 5-13, we compare the results of Scenario 1 with those obtained in the baseline case. The rejection probabilities reached almost 0.25 between 5 and 6 pm, compared to less than 0.18 in the baseline case during the same time of day.

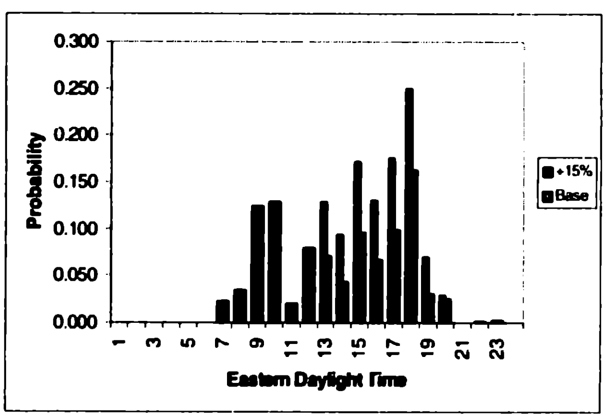


Figure 5-12: Scenario 1: New Rejection Probabilities vs. Baseline

We can see that the limited queue capacity has an important effect in regulating a controller's workload. Even though the demand increases considerably, the workload remains approximately the same. This effect of the limited queue can be seen as a "protection" on the controller's workload.

The average number of airplanes diverted during the day increased from 76.53 to 121.56, out of 1224 planes that crossed the sector. The total expected delay for the same 24 hours is 578.4 minutes, with all operations having an expected delay of less than a minute.

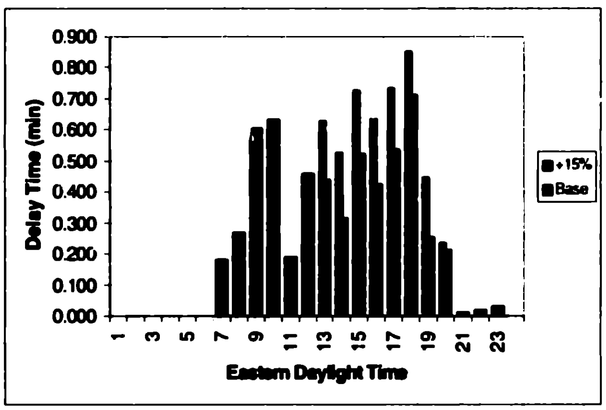


Figure 5-13: Scenario 1: New Expected Delays vs. Baseline

5.3.3 Scenario 2: Decreased Capacity in the Afternoon Hours

As discussed in Section 5.2.1, the sector capacity may vary during the day due to bad weather, severe turbulence or technical problems (e.g., radar or radio communication difficulties). In this example, we assume that a line of thunderstorms passes across the sector reducing its original capacity of 18 aircraft to the values shown in Table 5.1. The queueing model used for this example is the $M(t)/E_3/n(t)/n(t)+3$ system, where $n(t)$ is equal to 18 except for the hours shown in Table 5.1. We kept a constant service rate of $\mu = 4.2857$ aircraft per hour per server and the time-varying demand shown in Figure 5-3. The maximum hourly capacity was 77 aircraft, as in the baseline case,

Table 5.1: Reduced Capacity due to Bad Weather

Time of Day	2-3 pm	3-4 pm	4-5 pm	5-6 pm	6-7 pm	7-8 pm
Capacity (a/c)	15	12	9	9	12	15
Hourly Capacity (a/c)	64	51	38	38	51	64

except when the thunderstorms cross the sector. The system size is 472, 349, 244 and 157 states when the capacity of the sector is 18, 15, 12 and 9 aircraft, respectively. The required CPU time to compute the solution was 58 seconds.

Notice in Figure 5-14 that the expected number of airplanes in the sector was reduced to approximately 9 aircraft between 4 and 6 pm. To be able to compare the maximum controller workload in this scenario with that observed in the baseline case, we need to analyze the percentage difference between the expected workload and the maximum capacity at which controllers operate. In the baseline case, the maximum expected number of aircraft was 16.81, which is 6.5% below the maximum capacity of 18 airplanes. The corresponding maximum expected number of aircraft in the bad weather case, Scenario 2, during the same time of day, is 8.97 aircraft. This is a 0.3% difference from the maximum sector capacity of 9 airplanes. Clearly, this means a considerably higher level of utilization of the available capacity. According to our discussion in Section 5.2.1, the expected workload may also increase, as more planning, re-routing and monitoring is needed under poor weather conditions, even though the number of planes that passed through the sector decreased significantly.

Once again the effect of reducing the sector capacity is reflected in the rejection probabilities. As shown in Figure 5-15, the probability that a plane requesting to enter the sector will be rejected and re-routed increases to almost 0.60. Figure 5-15 presents the results for the hours when the reduction in capacity occurs. In this case, the average number of diverted airplanes during the day is 179.4, which is more than double that under normal conditions. In this case, the controller's workload is protected again by the limited queue capacity of the sector.

Due to the high congestion of the ZID095A sector between 5 and 6 pm, most

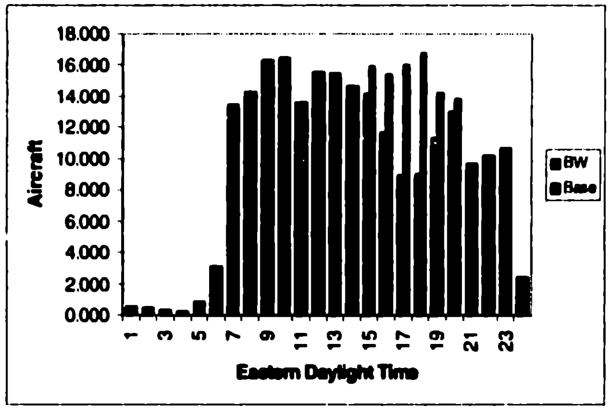


Figure 5-14: Scenario 2: New Expected Workload vs. Baseline

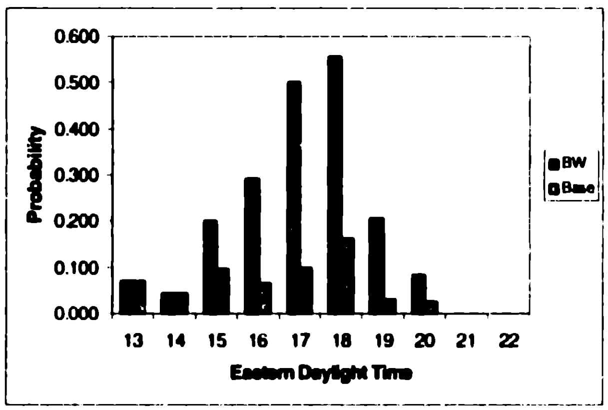


Figure 5-15: Scenario 2: New Rejection Probabilities vs. Baseline

aircraft that were granted access to the sector had to wait in the queue prior to entering the sector. Entering the queue means that an airplane has to reduce speed or has to vector in adjacent en-route sectors. Such operations increase the amount of delay of airplanes passing through the congested sector. Notice in Figure 5-16 that the expected delays increases considerably. Airplanes crossing the sector between 5 and 6 pm are delayed approximately 2 minutes, compared to less than a minute in the baseline case, and the total expected delay during the day increases from 430 minutes in the baseline case to 795.84 minutes.

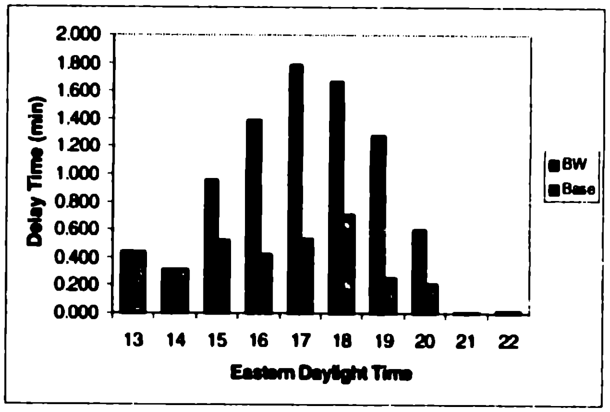


Figure 5-16: Scenario 2: New Expected Delays vs. Baseline

5.3.4 Scenario 3: Sensitivity of the Rejection Probability to Variations in the Queue Length

The purpose of this Scenario is to investigate the sensitivity of the saturation probabilities to changes in the capacity of the queue. The use of the assumption of limited queue size in modeling en-route sectors is an important one because not all requests of pilots to fly through a sector are granted. If the sector is highly congested and there are already as many airplanes as the controller can handle before they enter the sector, new requests to enter the sector would be denied. As we have seen before, the limited queue capacity has also an important effect on the controller's workload.

We use an $M(t)/E_3/18/18 + q$ queueing model with the values of q as shown in Table 5.2, column 1. The demand used is that of Figure 5-3 and the service rate is $\mu = 4.2857$ aircraft per hour per server, giving a maximum hourly capacity of 77 aircraft. The system size for the various values of q is also shown in Table 5.2, column 2. In this example, we focus only on the rejection probabilities.

Figure 5-17 shows the rejection probabilities during the day for numerous values of the queue size q. The maximum expected numbers of planes diverted in one hour are shown in Table 5.2. Notice that for $q = 50$, the sector behaves almost as if the queue had infinite capacity. A major consequence of increasing the queue capacity is that more airplanes are allowed to wait before entering the sector. Therefore, major

Table 5.2: System Size and Average Number of Aircraft Rejected as q Varies

q	System Size	Sum Over All Day	Maximum in One Hour
3	472	76.531	14.191
4	509	65.305	13.055
5	546	56.405	12.126
6	583	49.245	11.346
8	657	38.509	10.066
10	731	30.836	8.985
15	916	18.531	6.623
20	1101	11.237	4.611
30	1471	3.801	1.871
40	1841	1.132	0.638
50	2211	0.313	0.195

delays are expected to be experienced by airplanes that go through the sector.

Allowing an infinite queue capacity in an en-route sector is not a realistic consideration. The reason is that in most cases, airplanes would experience shorter delays if diverted to adjacent and less congested sectors. Intuitively, when the queue size

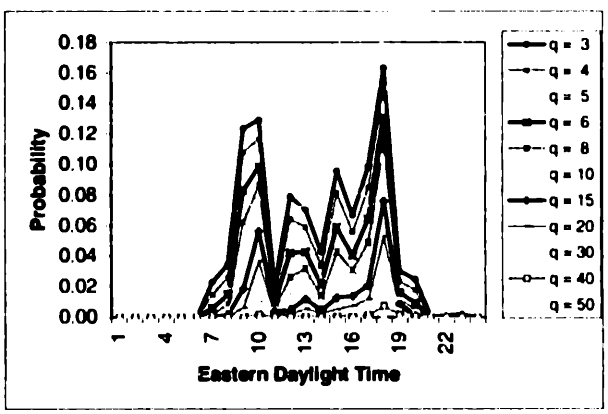


Figure 5-17: Scenario 3: Sensitivity of the Rejection Probabilities

is increased, the workload of controllers gets closer to the maximum sector capacity as more planes are constantly waiting to enter the en-route sector.

Table 5.3: Combined Sectors in a Four-Shift Pattern

Shift	Individual Sector Capacity	Number of Sectors Combined Under One Controller
1	6	3
2	18	1
3	18	1
4	12	1.5

5.3.5 Scenario 4: Combination of Multiple En-route Sectors

To conclude this Chapter, we present an example in which the capacity of the sector is reduced during low-traffic hours to combine two or more sectors into a single one. This is a current practice in air traffic control centers, as described in Section 5.2.1.

If we assume that there are 4 shifts of controllers during the day, each covering a period of 6 hours, we can combine the sectors as described in Table 5.3. Shift 1 starts at 12:01 am and ends at 6:00 am. Shifts 2 starts at 6:01 am, and so on. The first column in Table 5.3 indicates the shift number, the second column indicates the capacity of each individual sector, and the third column indicates the number of sectors assigned to each controller. For example under Shift 1, three original sectors have been combined into a single sector under one team of controllers. If the capacity of the team is to handle 18 aircraft simultaneously, then the capacity per original sector is 6. The model used in this example is an $M(t)/E_3/n(t)/n(t) + 3$ queueing model with the original demand of Figure 5-3, and a service rate of $\mu = 4.2857$ aircraft per hour per server. After combining the sectors in Shifts 1 and 4, the maximum hourly capacity of the combined sectors is 77 airplanes. The maximum hourly capacity of the single sectors during Shifts 2 and 3 is also 77 aircraft. The system sizes were 88, 472, 472 and 244 states for Shifts 1 through 4, respectively. In this experiment, the CPU time needed to obtain the solution was 40 seconds.

We present the results for all four shifts in Figure 5-18, the expected workload during the day, and Figure 5-19, the rejection probabilities. Shift 1 has such low demand that even after combining three sectors into one, the controller's workload always remains below 50% of its maximum capacity of 18 aircraft, and most of the time below 4.6% of the maximum capacity. The rejection probabilities are also influenced by the low demand. For example, most of the time in Shift 1, the sector behaves as if the queue would have infinite capacity, i.e., the rejection probabilities are zero from 12 am to 5 am. In Figure 5-18, the gray bars indicate the controller

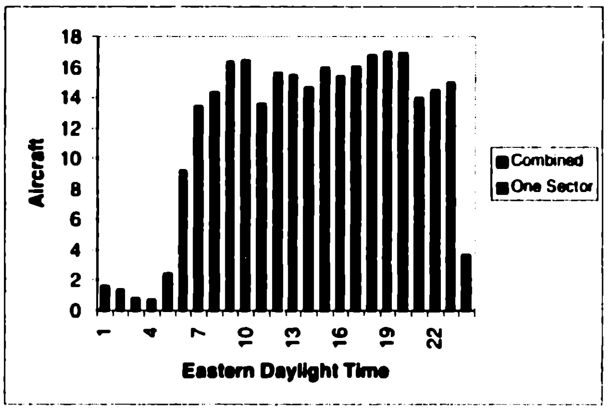


Figure 5-18: Scenario 4: Expected Workload of Air Traffic Controllers

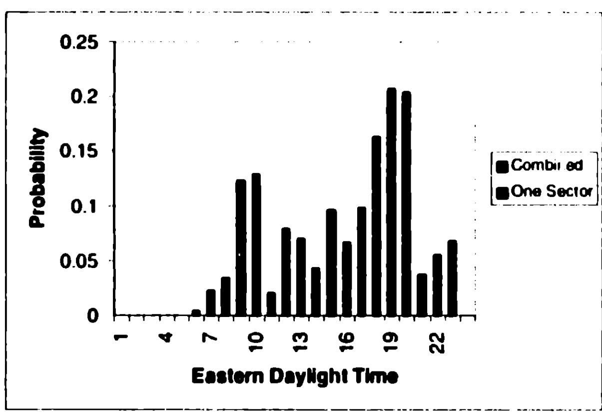


Figure 5-19: Scenario 4: Combined Rejection Probabilities

workload under normal operations. The black bars indicate the additional workload imposed on the controllers after combining multiple sectors into one. Figure 5-19 indicates in gray the rejection probabilities when the sectors are controlled separately, and indicates in black the rejection probabilities for Shifts 1 and 4, when multiple sectors are combined into one.

In the case of Shifts 2 and 3, we observe the same performance as in the baseline example. During those shifts, multiple sectors cannot be combined due to the high demand in the sector. During Shift 4, the expected controller's workload, as measured by utilization of maximum capacity, increases from 79% to 94.75% of the maximum capacity. The utilization is even higher than during the peak hour in the baseline case which is 93.4% of the maximum capacity. Regarding the rejection probabilities during Shift 4, the combination of multiple sectors causes those probabilities to increase to a maximum close to 0.21, compared to the corresponding maximum probability of 0.16 in the baseline case. A direct consequence of the increase in the rejection probabilities can be seen in the average number of planes that find a saturated queue. This number increases from 28.53 airplanes per day in the original case to 104.7 airplanes per day in the current example.

The analysis of the results for Shift 1 suggests that we may be able to assign more sectors to a single controller during that shift, while maintaining an adequate controller workload and low rejection probabilities. If we further modify Shift 1 and decrease the capacity to 3 airplanes, we are able to combine up to 6 sectors into one. In this case, we used an $M(t)/E_3/3/3+3$ queueing model to solve this experiment. The arrival information is that of Figure 5-3, and the service rate is $\mu = 4.2857$ airplanes per hour per server. During Shift 1, the system size reduces to 37 states and, after combining the 6 sectors, the maximum hourly capacity is 77 airplanes.

The expected workload for the modified Shift 1 example is exhibited in Figure 5-20. Notice that the expected workload for a controller increases from less than 50% to almost 83% of the maximum capacity between 5 and 6 am. This amount of workload is comparable to those experienced by controllers in charge of a single sector during Shifts 2 and 3. The rest of the time in Shift 1, the workload for the combined sectors is always below 27% of the maximum capacity. Figure 5-20 indicates with gray bars the controller workload for a single sector and with black bars the additional workload imposed on the air traffic controller after combining the six en-route sectors.

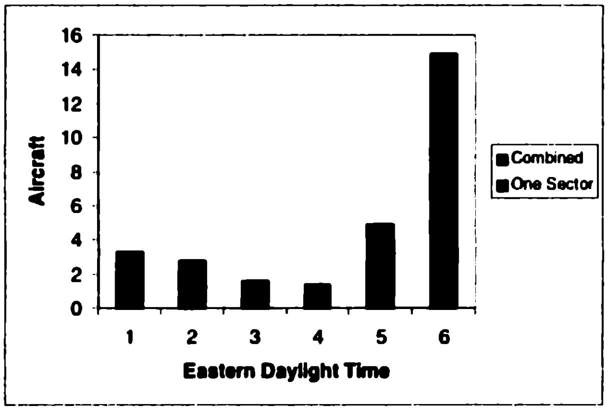


Figure 5-20: Scenario 4: Expected Workload for the Modified Shift 1

We also performed a sensitivity analysis of the rejection probabilities to changes in the number of servers. Our goal was to obtain the minimum value of $n(t)$, the capacity of the sector at time t, while maintaining the rejection probability below 0.10 at all times. In the cases where the probability is above 0.10, we do not modify the number of servers, i.e., we keep a maximum capacity of 18 airplanes in the sector. In this example, the system used was an $M(t)/E_3/n(t)/n(t)+3$ queue, where $n(t)$ was the variable in the sensitivity analysis. We used the arrival demand of Figure 5-3 and a service rate of $\mu = 4.2857$ aircraft per hour per server. To find the minimum values of n for the 24 hour period, we perform a trial and error approach. We started with the early hours of the day and continue until the end of the day. The minimum values of n are indicated in the third column of Table 5.4. In the same Table, the fourth column presents the system size for the various values of n. With the values of n in Table 5.4 the CPU time required to obtain the solutions was 55 seconds.

The rejection probabilities for the system with the minimum capacities are shown in Figure 5-21. The gray bars show the results obtained in the four-shift experiment and the black bars show the results using the minimum capacity values. Notice that most of the day, less than 10% of the requests to enter the sector are denied. Only during the peak demand hours, from 9 to 11 am and from 5 to 6 pm, the rejection probabilities are above 0.10 for the minimum capacities case.

Comparing the results of the two examples in Figure 5-21, we observe that the

Table 5.4: Minimum Values of n

From:	To:	Number of Servers per Sector	System Size
12:01 am	5:00 am	1	13
5:01 am	6:00 am	4	52
6:01 am	7:00 am	17	429
7:01 am	10:00 am	18	472
10:01 am	11:00 am	16	388
11:01 am	6:00 pm	18	472
6:01 pm	8:00 pm	17	429
8:01 pm	9:00 pm	12	244
9:01 pm	11:00 pm	13	277
11:01 pm	12:00 am	4	52

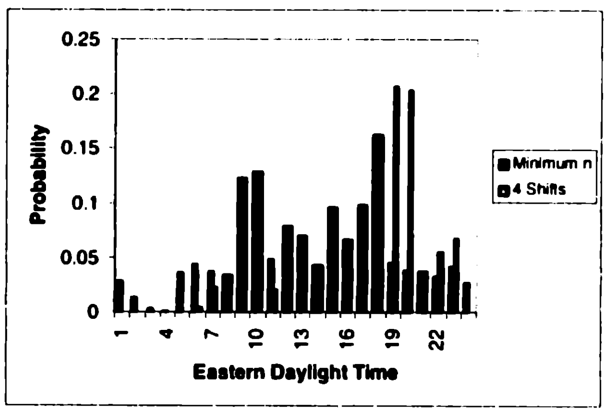


Figure 5-21: Scenario 4: Rejection Probabilities

rejection probabilities were reduced considerably between 6 and 8 pm as the capacity was increased form 12 to 17 aircraft. The capacities for both examples are shown in Figure 5-22. Similarly, the gray bars indicate the capacities for the four-shift example and the black bars indicate the minimum capacities. In Figure 5-22 we show the the minimum capacities and compare them to the four-shift capacities. Figures 5-21 and 5-22 suggest that an acceptable sector performance, in terms of rejection probabilities, can be achieved using the four-shift scenario. Notice that most of the day, the capacities in the four-shift scenario coincide with the minimum capacities of Table 5.4. Therefore, the rejection probabilities are close to those set by our

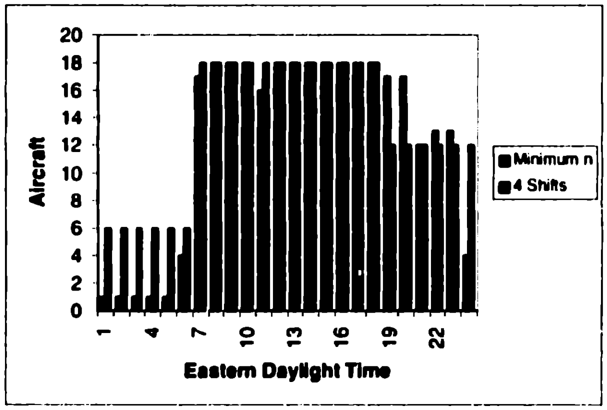


Figure 5-22: Scenario 4: Minimum Number of Servers

minimum acceptable level of service when obtaining the minimum capacities. As mentioned before, the major difference between both results occurs from 6 to 8 pm where the rejection probabilities in the four-shift scenario are considerably high. If we keep the proposed four controller shifts, we may be able to decrease the high rejection probabilities between 6 and 8 pm to the desired rejection probabilities of less than 0.10 by using some overtime of controllers from the 12:01-6:00 pm shift or, possibly, "back-up" controllers. Back-up controllers assist the actual controller assigned to one or more en-route sectors and help them to compensate for periods with large demands to assure acceptable rejection probabilities. Assisting controllers during high demand periods to process the traffic more efficiently is common practice in the air traffic control centers, as described in Section 5.2.1. A direct benefit of assisting controllers with either back-up controllers or controllers working overtime is reflected in savings for the FAA and airspace users. The FAA could assign fewer controllers during lower demand hours and may pay only for controllers' overtime and/or back-up controllers, and hence, reduce operating costs. On the other hand, airspace users could save money as they may incur fewer and shorter delays because traffic is processed faster and more efficiently during hours of high demand.

This scenario underlines the usefulness of the model with a variable number of servers. The capability of the model to vary the number of servers has been the key to balancing controller's workload and to effectively fulfilling staffing needs (including back-up controllers) for maintaining an adequate level of service in the sector, i.e., keeping the rejection probabilities below a certain threshold, and reducing costs of using and managing en-route sectors.

5.4 Summary

In this Chapter we addressed the implementation and usefulness of our heuristic solution technique. We have discussed an example of a possible application of the queueing model and our heuristic solution technique to an en-route ATC problem. Our case study for the high altitude sector shows that we can easily modify various system parameters and explore the effects of those changes using our heuristic solution technique.

Unfortunately, the lack of real data for actual capacities, delays and number of aircraft diverted from en-route sectors made it impossible to validate our case study results. Therefore, the scenarios presented in the case study are only indicative of the capabilities of the $M(t)/E_k(t)/n(t)/n(t)+q$ queueing systems to model en-route sectors and to evaluate their performance under various parameters.

If the $M(t)/E_k(t)/n(t)/n(t)+q$ model adequately portrays the behavior of en-route sectors, then this model can be a fast and easy-to-use tool that allows us to predict the workload, the level of service, the congestion and delays for en-route sectors. The use of this tool may be extremely helpful in the decision-making process at the strategic and policy level, as well as for the daily operations of en-route sectors. Some areas where the model may be of help are in determining sector boundaries, assigning controller shifts, fulfilling staffing needs, estimating delays, determining maximum capacities, and improving the overall air traffic flow in the airspace for present and future conditions of en-route sectors.

Chapter 6

Conclusions and Future Work

In this final Chapter, we first present the main conclusions of our research and state the theoretical and the practical contributions of the thesis. We then suggest potential areas of future research and describe possible ways to extend and apply our results.

6.1 Conclusions

We presented a thorough literature search and described numerous results available for $M(t)/G(t)/n/n+q$ queueing systems, especially for systems with Erlangian service time distributions. Our survey included systems under stationary and nonstationary conditions, and results for systems in the transient period, as well as for systems in steady-state. We observed several trends of research in queueing theory along the years and identified key results in the field that were the starting point of our work.

The main contribution of this thesis is an excellent approximation for certain $M(t)/G(t)/n(t)/n(t)+q$ queueing systems when the service time distribution is unimodal and has coefficient of variation less than one, using a heuristic solution technique for the $M(t)/E_k(t)/n(t)/n(t)+q$ queueing model. We developed a practical solution approach for the static and dynamic $M(t)/E_k(t)/n(t)/n(t)+q$ queueing model that is well structured, easily implemented and can be used in a wide variety of applications. We validated our solution technique with an extensive set of computational experiments involving queueing systems in steady-state and with stationary parameters and concluded that:

- The results of the ELC heuristic solution technique were always within 3% of the exact results, and 95% of the time the results were within 1% of the exact values.
- The size of the queueing systems that can be solved has been increased considerably using our heuristic solution approach: systems that were impossible to solve using the exact solution technique because of the large number of equations involved, were solved quickly using our heuristic solution technique.
- The time to solve the systems using our heuristic solution technique were up to 2,646 times faster than using the exact solution technique.

The performance of the ELC technique was excellent with a wide range of system parameters and appears to be quite robust to parameter changes.

An important feature included in our heuristic solution technique is the possibility of varying the number of servers in the model. We designed and tested an algorithm that maps the state probabilities of the systems before and after the number of servers changes. The results from the tests showed that our solution approach behaves intuitively well in modeling a change in the system capacity. With this feature, we are able to apply our heuristic solution technique to a wide range of realistic scenarios since many real-life problems involve variations in capacity in response to fluctuations in demand.

We also investigated the performance of our heuristic solution technique under nonstationary conditions and during the transient period from rest until the system reaches steady-state. The heuristic solution technique performed extremely well even under those circumstances. However, a larger set of examples is needed to fully validate our heuristic during the transient period and with dynamic parameters, including the case with a variable number of servers.

The last contribution of this thesis is the application presented in Chapter 5. We have provided an example of the usefulness of our model in real-life problems. The case study of the high altitude sector demonstrates the capabilities of the $M(t)/E_k(t)/n(t)/n(t)+q$ queueing system to model the behavior of en-route sectors allowing us to evaluate their performance under various scenarios of capacity and demand. We have also seen that our model can be used in the strategic planning process for maintaining adequate workload of air traffic controllers while at the same time assuring an acceptable level of service for en-route sector users. The modeling of an en-route sector presented in our work is a building block for a complete air traffic management tool for estimating capacity and delays in the air traffic management system. In the following Section, we provide some details concerning this potential Air Traffic Management modeling.

6.2 Future Work

Several lines of potential future research emerge from the results of this thesis. Some are related to the improved understanding of $M(t)/G(t)/n(t)/n(t)+q$ queueing systems and others to extending the application of such systems in Air Traffic Management.

We have identified two possible extensions in analyzing the behavior of multi-server systems with Poisson arrivals and general service time distributions:

1. Use the exact and/or ELC solution techniques to examine the transient period time constant in the $M/E_k/n/n+q$ queueing system. The analysis of the transient period can be done through an approach similar to the one used by Odoni and Roth [37] for the $M/M/1$ model.

2. Explore the use of different distributions for the service times, e.g. a combination of multiple Erlang distributions and/or phase-type distributions. To do this we need to be able to clearly identify the transition probabilities among states in the system, as well as the size of the system to solve. It is not obvious or straightforward that a combination of multiple Erlang distributions or of phase-type distributions may lead to feasible exact and/or heuristic so-

lution techniques. The usefulness of such service time distributions is that we would be able to capture a larger set of general service time distributions, e.g., multi-modal distributions, which are observed in several real-life problems. An example could be an en-route sector with two or more airways where most traffic is concentrated.

A direct extension of the application presented in Chapter 5 is to analyze several en-route sectors together, and include them in models such as AND and LMINET (see Chapter 2 for a description of those models) to build a complete Air Traffic Management tool. Within this new application, we found two important areas of research:

1. Account for the dependence among adjacent and non-adjacent en-route sectors.
2. Use ELC and, possibly, optimization tools to incorporate re-routing among en-route sectors.

The case study presented in Chapter 5 studied a single en-route sector without considering any interaction with other (adjacent or non-adjacent) en-route sectors. In order to adequately analyze several sectors together, we need to understand the interactions and interdependencies among en-route sectors. For example, if air traffic of an en-route Sector 1 comes mainly from a contiguous Sector 2, any alteration in the operations of Sector 2 would have a direct effect on the traffic that usually enters Sector 1. For instance, if Sector 2 experiences unusually high congestion or is affected by severe weather conditions reducing its capacity considerably, some of the following questions may arise regarding the operations in Sector 1: What would happen to the demand in Sector 1? Would Sector 1 experience higher or lower workload? Will the capacity of Sector 1 remain unchanged? Another example would be that if Sector 3 is adjacent to a highly utilized Sector 4, then if Sector 4 becomes saturated, Sector 3 may experience higher demand due to diverted airplanes from Sector 4.

Other interactions among sectors may be due to sectors that are not adjacent. To understand this situation, consider the following scenario. Flight F going from Boston, MA, to Atlanta, GA, passes through Sector A, covering part of Connecticut, and Sector B, covering the region to the East of the coast of North Carolina. If severe weather conditions affect the Northeast region of the U.S. and Sector A is diverting most of its traffic to the West, then Flight F may be re-routed to Sector C, covering part of New York State and Pennsylvania, and Sector D, covering parts of Ohio and West Virginia. Hence, the demands of Sectors B, C and D will be affected by problems in Sector A. It would be interesting to study and to understand the dependence of adjacent and non-adjacent en-route sectors, and to assess the degree of interaction among themselves.

The second extension to the application of Chapter 5 is very much related to the one just described. The problem of re-routing aircraft in the airspace involves several tasks and challenges. For each route between a pair of airports, we need to

- determine the alternate routes, along the complete flight trajectory, in case airplanes need to be diverted;
- know the weather, the demand and the capacity conditions for all sectors in the route and alternate routes;
- constantly update the status of all sectors in the original route and alternate routes;
- update the list of alternate routes as sector conditions may have changed;
- decide if aircraft have to be re-routed;
- efficiently select which aircraft are to be re-routed (minimize costs); and,
- optimally select the alternate route for diverted aircraft.

Determining the alternate routes requires only an exhaustive list of trajectories between the pairs of airports considered in the experiment. Knowing and updating sector conditions requires the use of our model to analyze all sectors that are included in the routes of the experiment, including the alternate ones. With the updated status for all en-route sectors, the list of alternate routes is updated and the decision to divert aircraft from saturated sectors is taken. The routes that include saturated sectors should be removed from the list of alternate routes as no aircraft can be re-routed through already saturated sectors. The decision to divert aircraft from saturated sectors may be based on the rejection probabilities obtained using our model.

Selecting the aircraft to be diverted and the new routes for them to follow is not straightforward. Some optimization tools may need to be used to minimize the costs for passengers, airlines and the FAA, and also to minimize the distance traveled, the delays incurred and the controllers' workload. A major challenge would be to obtain real data to validate the results. We faced that problem when trying to validate the results presented in our case study. If the re-routing problem is solved and en-route sectors are added to a model such as AND (see Chapter 2), along with re-routing capability, the result would be an extremely powerful and useful Air Traffic Management tool for modeling the complete air traffic system. Hence, this area of research includes many technical challenges and promises very interesting and practical applications.

Appendix A

States and State Transitions in the $M/E_4/3/4$ Queueing System

In this Appendix, we present, in detail, applications of the exact and heuristic solution techniques to the $M/E_4/3/4$ queue. The objective of this example is to illustrate the complexity of deriving the state transitions needed for the Chapman-Kolmogorov equations in even a small example. We also illustrate that the heuristic solution techniques are considerably simpler and easier to implement.

A.1 State Transitions: Exact Solution Technique

Figure A-1 is the state transition diagram for the $M/E_4/3/4$ queueing system, where the state is given by Description 2, (l, m, r), defined in Section 3.1.2. The associated transition rates are defined in Tables A.1, A.2 and A.3. The Chapman-Kolmogorov equations for the system can be obtained directly from these Tables. Note that the total number of states in Figure A-1 is given by Equation 3.1:

$$T_S = \begin{pmatrix} 3+4 \\ 3 \end{pmatrix} + 1 \times \begin{pmatrix} 3+4-1 \\ 3 \end{pmatrix} = 35 + 20 = 55.$$

Let us now explain the transitions described in Tables A.1 and A.2. Columns 1 through 5 contain the states according to Description 2, from Section 3.1.2. We

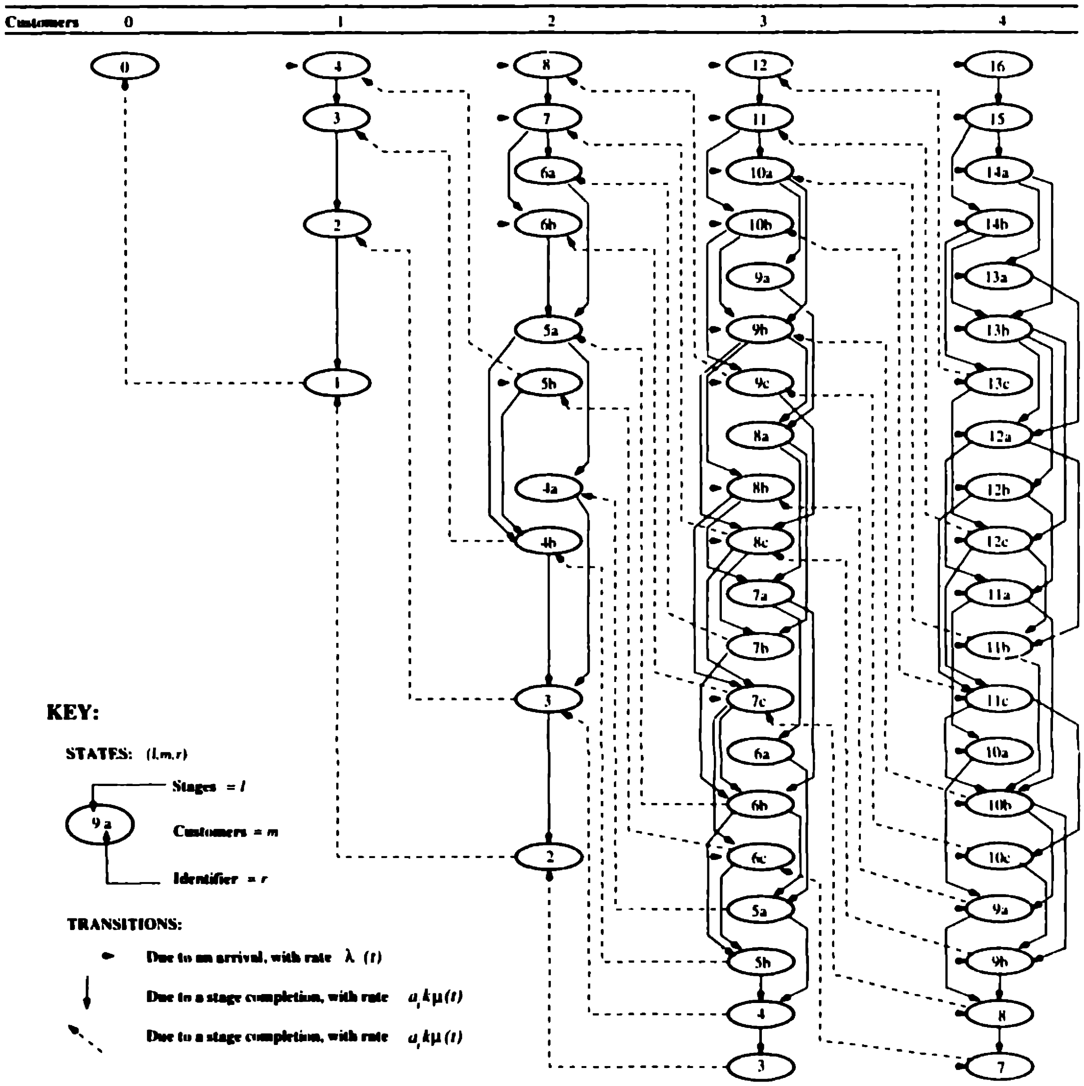


Figure A-1: $M/E_4/3/4$ queue, state transition diagram

can move out of a state in column 1 by either an arrival or a stage completion. If a customer enters the system, a transition from the state in column 1 to the state in column 2 occurs, with probability one. If a stage is completed, a transition from the state in column 1 to one of the states in column 3 occurs, and such transitions occur with the associated transition probability written at the right of the target state. For example, if a stage is completed while in state $(5,2,a)$ in column 1, we either go to state $(4,2,a)$, with probability $\frac{1}{2}$, or to state $(4,2,b)$, also with probability $\frac{1}{2}$.

We can enter a state in column 1 by either an arrival or a stage completion. If we enter the state in column 1 due to a customer arrival, the transition came from the state in column 4 with probability one. If we enter the state in column 1 due to a stage completion, a transition from one of the states in column 5 occurred, with the associated transition probability written at the right of the state, assuming that we were in that particular state. To better understand this last type of transition, consider the following example. If we entered state $(5, 2, a)$ because a stage was completed, the transition came from state $(6, 2, a)$ or state $(6, 2, b)$ or state $(6, 3, b)$. If the previous state was $(6, 2, a)$, the transition to state $(5, 2, a)$ occurred with probability one; if the system was in state $(6, 2, b)$, then the transition to state $(5, 2, a)$ occurred with probability $\frac{1}{2}$; finally, if the system was in state $(6, 3, b)$, then the transition to state $(5, 2, a)$ occurred with probability $\frac{1}{3}$.

The transitions shown in Table A.3 are similar to the transitions described above for Tables A.1 and A.2. The only difference is that in Table A.3 the transitions out of the states in column 1, due to a customer arrival, do not exist. The reason is that the system is saturated and no more customers can be accepted in the servers or in the queue. Therefore, we only have transitions out of the states in column 1, due to a stage completion, to the states in column 2. The transitions described in columns 2, 3 and 4 in Table A.3 are equivalent to the transitions in columns 3, 4 and 5 in Tables A.1 and A.2.

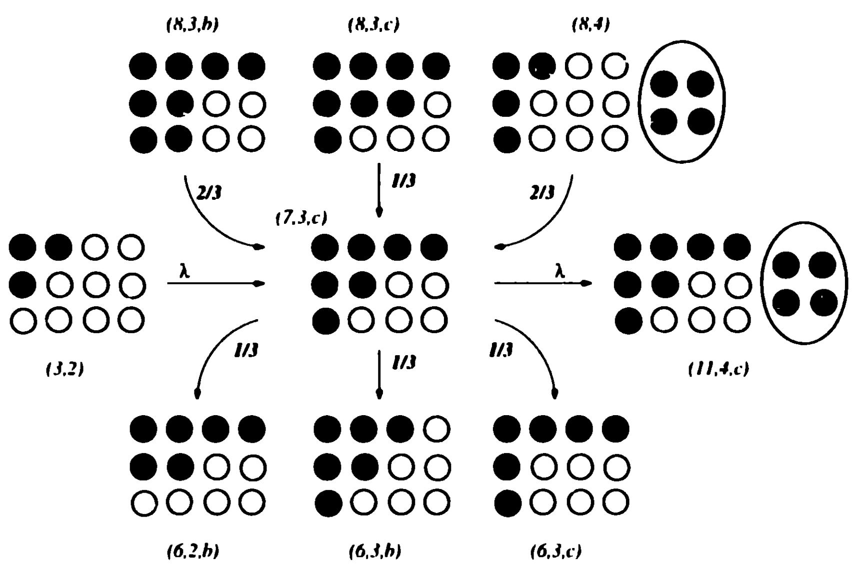


Figure A-2: Transitions in and out of state $(7, 3, c)$

In Figure A-2, all possible transitions for state $(7,3,c)$ are presented. The intention of this figure is to illustrate, in a more graphical way, how the transitions shown in Tables A.1, A.2 and A.3 occur. Recall that customers arrive at the system according to a Poisson process and that the time to complete a service stage is independent and exponentially distributed. There are four possible ways to enter state $(7,3,c)$ in any Δt:

1. from state $(3,2)$, when a customer arrives and enters directly to the available server, adding four uncompleted stages to the system;

2. from state $(8,3,b)$, with probability $\frac{2}{3}$, since two of the three servers in $(8,3,b)$ have two uncompleted stages; if any of those servers complete a stage, then we will have the same pattern as state $(7,3,c)$;

3. from state $(8,3,c)$, with probability $\frac{1}{3}$, if the server with three uncompleted stages finishes one of them; and,

4. from stage $(8,4)$, with probability $\frac{2}{3}$, if one of the two servers with one uncompleted stage completes that stage, the customer waiting in queue enters the freed server.

Note that in Figure A-2, a customer waiting in queue is represented by the four stages grouped in an oval.

Leaving state $(7,3,c)$, we have four cases as well. If a customer arrives, the transition is to state $(11,4,c)$, with the new customer waiting in queue until a server becomes available. If a stage is completed, three possible transitions may occur, each with probability $\frac{1}{3}$ since all servers are equally likely to complete a stage at a given time. The three transitions are as follows: to state $(6,2,b)$, leaving one server available; and to states $(6,3,b)$ and $(6,3,c)$ with all their servers busy.

A.2 State Transitions: Heuristic Solution Techniques

Note that the states with identical values of l and m are grouped in Tables A.1, A.2 and A.3. If we combine the states in each group, we would obtain a list of the states in the heuristic approximations. For example, in Figure A-3, we see how states $(7,3,a)$, $(7,3,b)$ and $(7,3,c)$ become state $(7,3)$. In the heuristic solution techniques, there

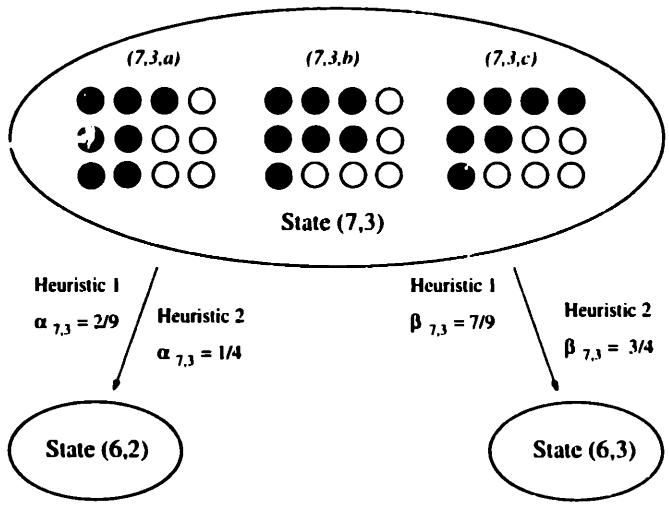


Figure A-3: Transitions out of state $(7,3)$ using either heuristic solution techniques

are two possible transitions out of state $(7,3)$, due to a stage completion. Under Heuristic 1, Equally Likely Patterns, the transition probabilities $\alpha_{7,3}$ and $\beta_{7,3}$ are obtained as follows:

Suppose that patterns a, b and c are all equally likely (probability $\frac{1}{3}$ each). The probability that a customer leaves the system, and thus the system will move from state $(7,3)$ to state $(6,2)$, is given by

$$\alpha_{7,3} = \left(\frac{1}{3} \times 0\right) + \left(\frac{1}{3} \times \frac{1}{3}\right) + \left(\frac{1}{3} \times \frac{1}{3}\right) = \frac{2}{9},$$

since pattern a has no server with only one uncompleted stage and both patterns b and c each have one out of three servers with only one uncompleted stage. If the transition is to a state with the same number of customers, i.e., to state $(6,3)$, the transition probability is

$$\beta_{7,3} = 1 - \alpha_{7,3} = \frac{7}{9}.$$

Let us now consider Heuristic 2, Equally Likely Combinations. We need to obtain the number of combinations for each pattern and the total for all three patterns. Patterns a and b each have two servers with the same number of uncompleted stages and a third server with a different number of uncompleted stages. Hence, both have the same number of combinations:

$$C_a = C_b = \frac{3!}{2!1!} = 3.$$

Pattern c has each of its three busy servers with a different number of uncompleted stages, and the number of combinations is

$$C_c = \frac{3!}{1!1!1!} = 6.$$

Therefore, the total number of combinations in state $(7,3)$ is given by

$$C_{total} = C_a + C_b + C_c = 12.$$

The probability of patterns a and b are $\frac{3}{12} = \frac{1}{4}$ each, and the probability of pattern c is $\frac{6}{12} = \frac{1}{2}$. Hence, the transition probability from state $(7,3)$ to state $(6,2)$ is given by

$$\alpha_{7,3} = \left(\frac{1}{4} \times 0\right) + \left(\frac{1}{4} \times \frac{1}{3}\right) + \left(\frac{1}{2} \times \frac{1}{3}\right) = \frac{1}{4},$$

with pattern a having no customers with one uncompleted stage, and patterns b and c having one of three servers with only one uncompleted stage. The transition probability from state $(7,3)$ to state $(6,3)$ is thus

$$\beta_{7,3} = 1 - \alpha_{7,3} = \frac{3}{4}.$$

Finally, the simplified state transition diagram for the heuristic solution techniques is presented in Figure A-4. Note that even for a small system, like the $M/E_4/3/4$ queue, the heuristics simplify considerably the transitions between states. The transition probabilities $\alpha_{l,m}$ and $\beta_{l,m}$ are obtained with Equations 3.11, 3.12 and 3.15, in Section 3.1.4. Once the state transition probabilities have been obtained, the Chapman-Kolmogorov equations may be derived using Equations 3.19 through 3.30. The total number of Chapman-Kolmogorov equations which corresponds to the state

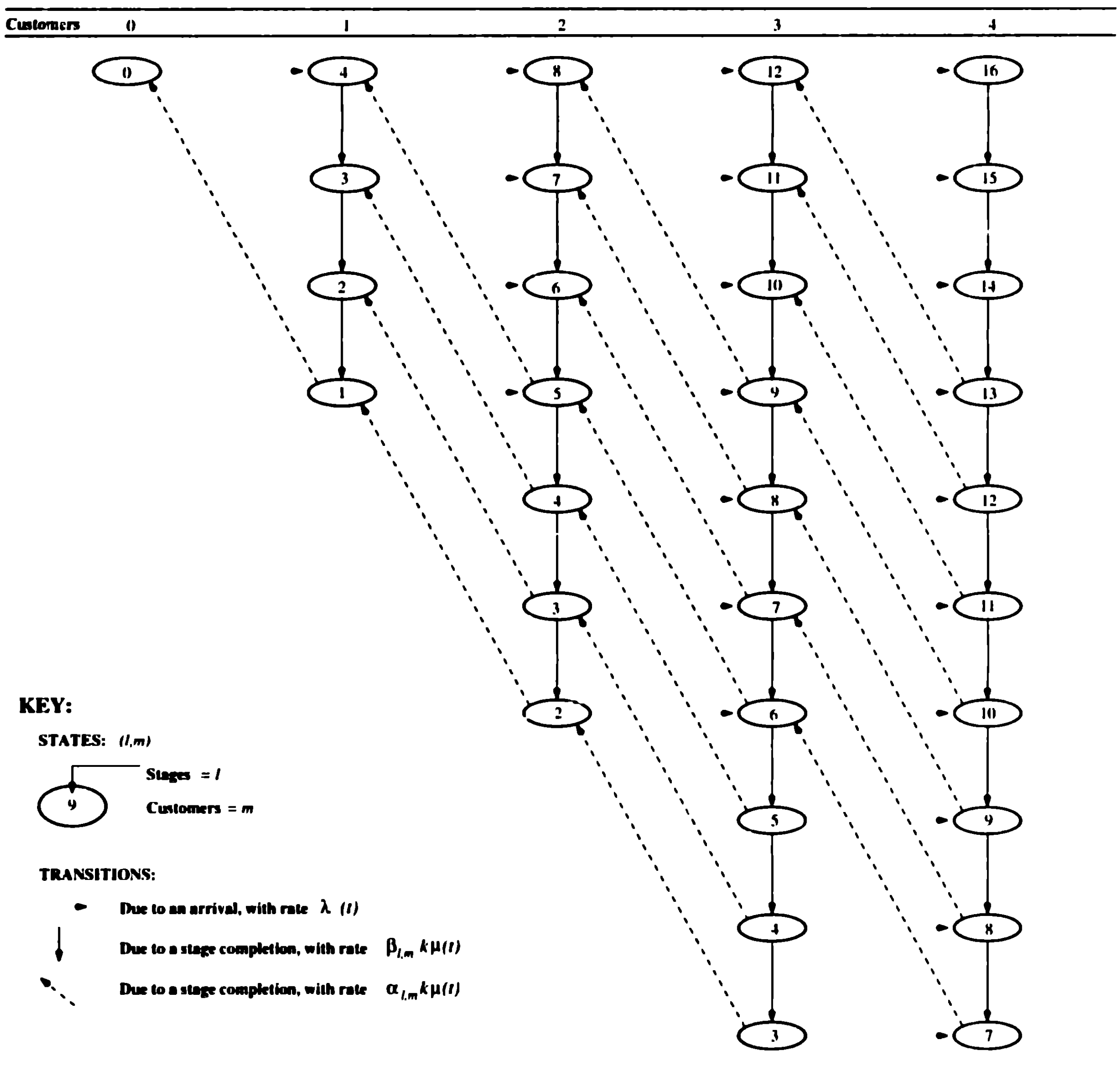


Figure A-4: $M/E_4/3/4$ queue, simplified state transition diagram

transition diagram of Figure A-4 is given by

$$S_S = (4-1)\left[\frac{3(3+1)}{2} + 1 \times 3\right] + 1 + 3 + 1 = 3(9) + 5 = 32,$$

from Equation 3.18.

With this example, we have shown how the state-to-state transitions occur in both the exact solution technique and the heuristic solution techniques. Numerical results of this example for selected performance measures are presented in Section 4.1.

Table A.1: State Transitions. Part 1: 0, 1 and 2 Customers in the System

State (l,m,p,r)	To State (Arrival)	To State, Prob. (Stage Completion)	From State (Arrival)	From State, Prob. (Stage Completion)
(0,0)	(4,1)	—	—	(1,1),1
(4,1)	(8,2)	(3,1),1	(0,0)	(5,2,b),1/2
(3,1)	(7,2)	(2,1),1	—	(4,1),1; (4,2,b),1/2
(2,1)	(6,2,b)	(1,1),1	—	(3,1),1; (3,2),1/2
(1,1)	(5,2,b)	(0,0),1	—	(2,1),1; (2,2),1
(8,2)	(12,3)	(7,2),1	(4,1)	(9,3,c),1/3
(7,2)	(11,3)	(6,2,a),1/2; (6,2,b),1/2	(3,1)	(8,2),1; (8,3,c),1/3
(6,2,a)	(10,3,a)	(5,2,a),1	—	(7,2),1/2; (7,3,b),1/3
(6,2,b)	(10,3,b)	(5,2,a),1/2; (5,2,b),1/2	(2,1)	(7,2),1/2; (7,3,c),1/3
(5,2,a)	(9,3,b)	(4,2,a),1/2; (4,2,b),1/2	—	(6,2,a),1; (6,2,b),1/2
	—	—	—	(6,3,b),1/3
(5,2,b)	(9,3,c)	(4,1),1/2; (4,2,b),1/2	(1,1)	(6,2,b),1/2; (6,3,c),2/3
(4,2,a)	(8,3,b)	(3,2),1	—	(5,2,a),1/2; (5,3,a),1/3
(4,2,b)	(8,3,c)	(3,1),1/2; (3,2),1/2	—	(5,2,a),1/2; (5,2,b),1/2
	—	—	—	(5,3,b),2/3
(3,2)	(7,3,c)	(2,1),1/2; (2,2),1/2	—	(4,2,a),1; (4,2,b),1/2
	—	—	—	(4,3),2/3
(2,2)	(6,3,c)	(1,1),1	—	(3,2),1/2; (3,3),1

Table A.2: State Transitions. Part 2: 3 Customers in the System

State (l,m,p,r)	**To State** (Arrival)	**To State, Prob.** (Stage Completion)	**From State** (Arrival)	**From State, Prob.** (Stage Completion)
(12,3)	(16,4)	(11,3),1	(8,2)	(13,4,c),1/3
(11,3)	(15,4)	(10,3,a),2/3; (10,3,b),1/3	(7,2)	(12,3),1; (12,4,c),1/3
(10,3,a)	(14,4,a)	(9,3,a),1/3; (9,3,b),2/3	(6,2,a)	(11,3),2/3; (11,4,b),1/3
(10,3,b)	(14,4,b)	(9,3,b),2/3; (9,3,c),1/3	(6,2,b)	(11,3),1/3; (11,4,c),1/3
(9,3,a)	(13,4,a)	(8,3,a),1	—	(10,3,a),1/3
(9,3,b)	(13,4,b) —	(8,3,a),1/3; (8,3,b),1/3; (8,3,c),1/3	(5,2,a) —	(10,3,a),2/3; (10,3,b),2/3; (10,4,b),1/3
(9,3,c)	(13,4,c)	(8,2),1/3; (8,3,c),2/3	(5,2,b)	(10,3,b),1/3; (10,4,c),2/3
(8,3,a)	(12,4,a)	(7,3,a),2/3; (7,3,b),1/3	—	(9,3,a),1; (9,3,b),1/3
(8,3,b)	(12,4,b)	(7,3,a),1/3; (7,3,c),2/3	(4,2,a)	(9,3,b),1/3; (9,4,a),1/3
(8,3,c)	(12,4,c) —	(7,3,b),1/3; (7,3,c),1/3 (7,2),1/3	(4,2,b) —	(9,3,b),1/3; (9,3,c),2/3; (9,4,b),2/3
(7,3,a)	(11,4,a)	(6,3,a),1/3; (6,3,b),2/3	—	(8,3,a),2/3; (8,3,b),1/3
(7,3,b)	(11,4,b)	(6,3,b),2/3; (6,2,a),1/3	—	(8,3,a),1/3; (8,3,c),1/3
(7,3,c)	(11,4,c) —	(6,3,b),1/3; (6,3,c),1/3; (6,2,b),1/3	(3,2) —	(8,3,b),2/3; (8,3,c),1/3; (8,4),2/3
(6,3,a)	(10,4,a)	(5,3,a),1	—	(7,3,a),1/3
(6,3,b)	(10,4,b) —	(5,3,a),1/3; (5,3,b),1/3; (5,2,a),1/3	— —	(7,3,a),2/3; (7,3,b),2/3; (7,3,c),1/3
(6,3,c)	(10,4,c)	(5,3,b),1/3; (5,2,b),2/3	(2,2)	(7,3,c),1/3; (7,4),1
(5,3,a)	(9,4,a)	(4,3),2/3; (4,2,a),1/3	—	(6,3,a),1; (6,3,b),1/3
(5,3,b)	(9,4,b)	(4,3),1/3; (4,2,b),2/3	—	(6,3,b),1/3; (6,3,c),1/3
(4,3)	(8,4)	(3,3),1/3; (3,2),2/3	—	(5,3,a),2/3; (5,3,b),1/3
(3,3)	(7,4)	(2,2),1	—	(4,3),1/3

Table A.3: State Transitions. Part 3: 4 Customers in the System

State (l,m,p,r)	**To State, Prob.** (Stage Completion)	**From State** (Arrival)	**From State, Prob.** (Stage Completion)
(16,4)	(15,4),1	(12,3)	—
(15,4)	(14,4,a),2/3; (14,4,b),1/3	(11,3)	(16,4),1
(14,4,a)	(13,4,a),1/3; (13,4,b),2/3	(10,3,a)	(15,4),2/3
(14,4,b)	(13,4,b),2/3; (13,4,c),1/3	(10,3,b)	(15,4),1/3
(13,4,a)	(12,4,a),1	(9,3,a)	(14,4,a),1/3
(13,4,b)	(12,4,a),1/3; (12,4,b),1/3;	(9,3,b)	(14,4,a),2/3; (14,4,b),2/3
	(12,4,c),1/3	—	—
(13,4,c)	(12,4,c),2/3; (12,3),1/3	(9,3,c)	(14,4,b),1/3
(12,4,a)	(11,4,a),2/3; (11,4,b),1/3	(8,3,a)	(13,4,a),1; (13,4,b),1/3
(12,4,b)	(11,4,a),1/3; (11,4,c),2/3	(8,3,b)	(13,4,b),1/3
(12,4,c)	(11,4,b),1/3; (11,4,c),1/3;	(8,3,c)	(13,4,b),1/3; (13,4,c),2/3
	(11,3),1/3	—	—
(11,4,a)	(10,4,a),1/3; (10,4,b),2/3	(7,3,a)	(12,4,a),2/3; (12,4,b),1/3
(11,4,b)	(10,4,b),2/3; (10,3,a),1/3	(7,3,b)	(12,4,a),1/3; (12,4,c),1/3
(11,4,c)	(10,4,b),1/3; (10,4,c),1/3;	(7,3,c)	(12,4,b),2/3; (12,4,c),1/3
	(10,3,b),1/3	—	—
(10,4,a)	(9,4,a),1	(6,3,a)	(11,4,a),1/3
(10,4,b)	(9,4,a),1/3; (9,4,b),1/3;	(6,3,b)	(11,4,a),2/3; (11,4,b),2/3
	(9,3,b),1/3	—	(11,4,c),1/3
(10,4,c)	(9,4,b),1/3; (9,3,c),2/3	(6,3,c)	(11,4,c),1/3
(9,4,a)	(8,4),2/3; (8,3,b),1/3	(5,3,a)	(10,4,a),1; (10,4,b),1/3
(9,4,b)	(8,4),1/3; (8,3,c),2/3	(5,3,b)	(10,4,b),1/3; (10,4,c),1/3
(8,4)	(7,4),1/3; (7,3,c),2/3	(4,3)	(9,4,a),2/3; (9,4,b),1/3
(7,4)	(6,3,c),1	(3,3)	(8,4),1/3

Bibliography

[1] Federal Aviation Administration. FAA Order 7210.3M, Change 1, May 1996. U.S. Department of Transportation, Washington, DC.

[2] D. Bertsimas and S. Stock. The Air Traffic Flow Management Problem with Enroute Capacities. *Unpublished*, 1994.

[3] O.J. Boxma, J.W. Cohen, and N. Huffels. Approximations of the Mean Waiting Time in an $M/G/s$ Queueing System. *Operations Rcsearch*, 1979. Vol. 27, pp. 1115-1127.

[4] G.P. Cosmetatos. Some Approximate Equilibirium Results for the Multi-server Queue $(M/G/r)$. *Operational Research Quarterly*, 1976. Vol. 27, pp. 615-620.

[5] L.V. Green and P.J. Kolesar. The Pointwise Stationary Approximation for Queues with Nonstationary Arrivals. *Management Science*, 1991. Vol. 37, No. 1, pp. 84-97.

[6] L.V. Green and P.J. Kolesar. On the Accuracy of the Simple Peak Hour Approximation for Markovian Queues. *Management Science*, 1993. Vol. 39.

[7] L.V. Green, P.J. Kolesar, and A. Svoronos. Some Effects of Nonstationarity on Multi-Server Markovian Systems. *Operations Research*, 1991. Vol. 39, No. 3, pp. 502-511.

[8] D. Gross and C.M. Harris. *Fundamentals of Queueing Theory*. John Wiley and Sons, 1974.

[9] D.P. Heyman and W. Whitt. The Asymptotic Behavior of Queues with Time-Varying Arrival Rates. *Journal of Applied Probability*, 1984. Vol. 21, pp. 143-156.

[10] F.S. Hillier and F.D. Lo. Tables for Multiple-Server Queueing Systems Involving Erlang Distributions, 1971. Technical Report No. 31, Department of Operations Research, Stanford University.

[11] P. Hokstad. Approximations for the $M/G/m$ Queue. *Operations Research*, 1978. Vol. 26, No. 3, pp. 510-523.

[12] W.D. Kelton and A.M. Law. The Transient Behavior of the $M/M/s$ Queue, with Implications for Steady-State Simulation. *Operations Research*, 1985. Vol. 33, No. 2, pp. 378-396.

[13] T. Kimura. Diffusion Approximation for an $M/G/s$ Queue. *Operations Research*, 1983. Vol. 31, No. 2, pp. 304-319.

[14] T. Kimura. A Transform-Free Approximation for the Finite Capacity $M/G/s$ Queue. *Operations Research*, 1996. Vol. 44, No. 6, pp. 984-988.

[15] J.F.C. Kingman. Inequalities in the Theory of Queues. *J. R. Stat. Soc.*, 1970.

[16] P. Kivestu. *Alternative Methods of Investigating the Time-Dependent $M/G/k$ Queue*, 1976. Massachusetts Institute of Technology, SM Thesis.

[17] L. Kleinrock. *Queueing Systems, Volume 1: Theory*. John Wiley and Sons, 1975.

[18] L. Kleinrock. *Queueing Systems, Volume 2: Theory*. John Wiley and Sons, 1975.

[19] J. Kollerstrom. Heavy Traffic Theory for Queues with Several Servers. *Applied Probability*, 1974.

[20] B. Koopman. Air Terminal Queues Under Time-Dependent Conditions. *Operations Research*, 1972. Vol. 20, pp. 1089-1114.

[21] D. Lee. $M/E_3/N/N+Q$ and $M/E_K/N/N+Q$, January 1997. LMI Internal Paper.

[22] D. Lee, P. Kostiuk, B. Kaplan, M. Escobar, A. Odoni, and B. Malone. Technical and Economic Analysis of Air Transportation Mangement Issues Related to Free Flight, February 1997. Logistics Management Institute, Technical Report NS501T1.

[23] D. Lee, D. Long, M. Etheridge, J. Plugge, J. Johnson, and P. Kostiuk. A Method for Making Cross-comparable Estimates of the Benefits of Decision Support Technologies for Air Traffic Management, October 1997. Logistics Management Institute, Technical Report NS710.

[24] B. Ma and J.W. Mark. Approxin ' of the Mean Queue Length of an $M/G/c$ Queueing Sytem. *Operations R* 1995. Vol. 43, No. 1, pp. 158-165.

[25] E. Maaløe. Approximation Formulae for Estimation of Waiting-time in Multiple-Channel Queueing Systems. *Management Science*, 1973. Vol. 19, pp. 703-710.

[26] K. Malone. *Modeling a Network of Queues Under Nonstationary and Stochastic Conditions*, 1993. Massachusetts Institute of Technology, SM Thesis.

[27] K. Malone. *Dynamic Queueing Systems: Behavior and Approximations for Individual Queues and for Networks.* PhD thesis, Massachusetts Institute of Technology, June 1995.

[28] J.O. Mayhugh and R.E. McCormick. Steady State Solution of the Queue $M/E_k/r$. *Management Science*, 1968. Vol. 14, pp. 692-712.

[29] M. Miyazawa. Approximation of Queue-Length Distribution of an $M/GI/s$ Queue by the Basic Equations. *Journal of Applied Probability*, 1986. Vol. 23, pp. 443-458.

[30] J.R. Murray and W.D. Kelton. The Transient Behavior of the $M/E_k/2$ Queue in Steady-State Simulation. *Computers and Operations Research*, 1988. Vol. 15, No. 4, pp. 357-367.

[31] G.F. Newell. Queues with Time-Dependent Arrival Rates I-III. *Journal of Applied Probability*, 1968. Vol. 5, pp. 436-606.

[32] G.F. Newell. Approximate Stochastic Behavior of n-Server Service Systems with Large n, 1973. Lecture Notes in Economics and Math. Systems, Vol. 87, Springer-Verlag.

[33] S.A. Nozaki and S.M. Ross. Approximations in Finite-Capacity Multi-Server Queues with Poisson Arrivals. *Journal of Applied Probability*, 1978. Vol. 15, pp. 826-834.

[34] A.R. Odoni. Issues in Air Traffic Flow Management, 1991.

[35] A.R. Odoni, J. Bowman, D. Delahaye, J. Deyst, E. Feron, R.J. Hansman, K. Khan, J.K. Kuchar, N. Pujet, and R. Simpson. Modeling Research Under NASA/AATT. Final Report. Existing and Required Modeling Capabilities for Evaluating ATM Systems and Concepts, March 1997. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

[36] A.R. Odoni and P. Kivestu. A Handbook for the Estimation of Airside Delays at Major Airports (Quick Approximation Method). *NASA Contractor Report*, 1976. CR 2644.

[37] A.R. Odoni and E. Roth. An Empirical Investigation of the Transient Behavior of Stationary Queueing Systems. *Operations Research*, 1983. Vol. 31, No. 3, pp. 432-455.

[38] O. Richetta and A.R. Odoni. Solving Optimally the Static Ground-Holding Policy Problem in Air Traffic Control. *Transportation Science*, 1993. Vol. 27, No. 3.

[39] O. Richetta and A.R. Odoni. Dynamic Solution to the Ground-Holding Problem in Air Traffic Control. *Transportation Research*, 1994. Vol. 28A, No. 3.

[40] S. Shapiro. The M-Server Queue with Poisson Input and Gamma-Distributed Service of Order Two. *Journal of the Operations Research Society of America*, 1966. Vol. 14, pp. 685-694.

[41] J.H. De Smit. Some General Results for Many Server Queues. *Advances in Applied Probability*, 1973. Vol. 5, pp. 153-169.

[42] V.L. Smith. Distribution of Customers in $M/E_T/m$ Queues using Hokstad's Approximation. *Applied Statistics (Royal Stat. Soc.)*, 1987. Vol. 36, pp. 394-401.

[43] D. Stoyan. Some Bounds for Many-Server Systems $Gl/G/s$. *Math. Operat.-forshung Statist.*, 1974.

[44] Y. Takahashi. An Approximation Formula for the Mean Waiting Time of an $M/G/c$ Queue. *Journal of the Operations Research Society of Japan*, 1977. Vol. 20, No. 3, pp. 150-163.

[45] H.C. Tijms, M.H. van Hoorn, and A. Federgruen. Approximations for the Steady-State Probabilities in the $M/G/c$ Queue. *Advance Applied Probability*, 1981. Vol. 13, pp. 186-206.

[46] W. Whitt. The Pointwise Stationary Approximation for $M_t/M_t/s$ Queues is Asymptotically Correct as the Rates Increase. *Management Science*, 1991. Vol. 37, No. 3, pp. 307-315.

[47] H. Wilhelmsen, C.W. Crone, T.A. Petrillo Jr., M.E. Hanson, J.R. Eggert, and V. Heinz. Air Traffic Control Overview: Kansas City ARTCC, January 1998. MIT Lincoln Laboratory.

[48] R.W. Wolff. Poisson Arrivals See Time Averages. *Operations Research*, 1982. Vol. 30, No. 2, pp. 223-231.

[49] D.D. Yao. Refining the Diffusion Approximation for the $M/G/m$ Queue. *Operations Research*, 1985. Vol. 33, pp. 1266-1277.